



Universitat Autònoma
de Barcelona

Consultes a Base de Dades en Llenguatge Natural

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes

realitzat per

Raimon Izard Serrano

i dirigit per

Jordi Duran Cals

Escola d'Enginyeria

Sabadell, *Juny* de 2010

El sotasignat, **Jordi Duran i Cals**,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en **Raimon Izard i Serrano**

I per a que consti firma la present.
Sabadell, **Juny** de **2010**

Signat: **Jordi Duran i Cals**

Resum del projecte

El present document relata, especifica i justifica l'elaboració del Projecte Final de Carrera de la diplomatura d'Enginyeria Tècnica d'Informàtica de Sistemes defensat per l'alumne Raimon Izard Serrano a l'*Escola Universitària d'Informàtica de Sabadell – EUIS* al Juny del 2010.

El treball presenta un prototip que ofereix la possibilitat d'obtenir informació d'una base de dades usant consultes en Llenguatge Natural a través d'un entorn web distribuït.

La idea sorgeix directament del món laboral gràcies a la meva estada a l'empresa *Accenture Consulting* en conveni de becari amb la *UAB*. Allà m'adono de la necessitat d'un sistema o eina que pugui subministrar l'accés a la informació continguda a la base de dades per part dels treballadors no tècnics ni experts en entorns de bases de dades ni tampoc en sentències SQL.

Amb el nostre prototip hem complert l'objectiu principal que és el de subministrar accés a la informació a usuaris no experts de manera àgil i senzilla. Amb l'afegit d'englobar-ho tot en un entorn web distribuït que estalvia les instal·lacions als equips dels usuaris finals, ofereix integració complerta dins l'entorn de treball existent i la possibilitat de consultar l'estat de les peticions al nostre sistema des de qualsevol punt.

El llenguatge és un procés de creació lliure, les seves lleis i principis són fixos, però la manera en què els principis de la generació s'utilitzen és lliure i infinitament variada. Fins i tot la interpretació i l'ús de paraules implica un procés de creació lliure.

Noam Chomsky

Taula de continguts

1.	Introducció.....	7
1.1.	Motivacions	7
1.2.	Estructura de la memòria del projecte	7
2.	Estudi de viabilitat	9
2.1.	Introducció.....	9
2.2.	Tipologia i paraules clau.	9
2.2.1.	Tipologia:.....	9
2.2.2.	Paraules Clau:.....	9
2.3.	Descripció.	10
2.4.	Objectius del projecte.	12
2.4.1.	Llistat d'objectius.	13
2.4.2.	Catalogació dels objectius.....	13
2.5.	Parts interessades.	14
2.5.1.	Stakeholders	14
2.5.2.	Perfils d'usuari.....	14
2.5.3.	Equip de projecte	15
2.6.	Producte i documentació del projecte.....	15
2.7.	Estudi de la situació actual	16
2.8.	Context.....	18
2.9.	Lògica del sistema.	19
2.10.	Descripció física.....	20
2.11.	Usuaris i/o personal del sistema	21
2.12.	Diagnòstic del sistema	22
2.13.	Normatives i legislació.	23
2.14.	Requisits del projecte.....	24
2.14.1.	Llistat de requisits funcionals.....	24
2.14.2.	Catalogació dels requeriments funcionals	24
2.14.3.	Llistat de requisits no funcionals.....	25
2.14.4.	Catalogació dels requeriments no funcionals	25
2.15.	Restriccions del sistema.	26
2.16.	Catalogació i prioritització dels requisits.....	26
2.17.	Alternatives i selecció de la solució	27
2.17.1.	Alternativa 1: SQ-HAL: Natural Language to SQL Translator.....	27
2.17.2.	Alternativa 2: English Query 2000	28
2.17.3.	Solució proposada.....	29
2.18.	Planificació del projecte	30
2.19.	Recursos del projecte.	31
2.19.1.	Recursos humans	31
2.19.2.	Recursos materials.....	32
2.20.	Tasques del projecte.....	34
2.21.	Planificació temporal.....	37
2.22.	Avaluació de riscos.....	38
2.22.1.	Llista de riscos.....	38
2.22.2.	Catalogació de riscos	38
2.23.	Pla de contingència.....	39
2.24.	Pressupost	40
2.24.1.	Estimació cost de personal.....	40
2.24.2.	Estimació cost dels recursos.	41
2.24.3.	Estimació cost de les activitats	42
2.24.4.	Estimació d'altres costos.....	44
2.24.5.	Resum i anàlisi cost benefici.	44
2.25.	Conclusions	45
3.	Disseny Tècnic i implementació del Sistema	46
3.1.	Model de desenvolupament.....	46
3.2.	Entorn de desenvolupament.....	46
3.3.	Disseny del prototip	49
3.4.	Disseny i implementació del flux de processament.....	52
3.4.1.	Disseny de la gramàtica	53
3.5.	Disseny de la Base de Dades.....	56
3.5.1.	Base de dades del sistema	56

3.5.2.	Base de dades del prototip.....	60
3.5.3.	Disseny d'adaptació de la BD del client	61
3.6.	Disseny de l'aplicació web	64
3.7.	Gestor de consultes	70
3.8.	Diagrames de seqüència, aplicació i casos d'ús	72
4.	Fase de proves	83
4.1.	Proves d'unitat de l'analitzador del llenguatge i la gramàtica	83
4.2.	Proves d'unitat de la BD	84
4.3.	Proves d'unitat de l'aplicació web	85
4.4.	Proves d'unitat del gestor de consultes	87
5.	Conclusions.....	88
5.1.	Valoració d'objectius assolits	88
5.2.	Futures línies de treball.....	90
5.3.	Seguiment de la planificació	91
5.4.	Valoració personal	98
6.	Glossari	100
6.1.	Acrònims i abreviacions	100
6.2.	Relació de taules	102
6.3.	Relació de figures i il·lustracions.....	103
7.	Bibliografia	104
7.1.	Escrita	104
7.1.1.	Apunts	104
7.1.2.	Llibres.....	104
7.2.	Digital	105
8.	Agraïments	106

1. Introducció

1.1. *Motivacions*

Partim de la idea de resoldre una necessitat real dins del món laboral, aleshores el projecte esdevé molt ambiciós i motivant perquè d'entrada no renunciem a res. Simplement fem una sessió de pluja d'idees per dissenyar un producte teòric capaç de solucionar la mancança de la millor manera possible en un únic sistema integrat. A més a més, s'escull un llenguatge de programació de nova generació tal com el *Ruby on Rails* i les seves llibreries amb l'objectiu d'obtenir un prototip modern que miri cap el futur per tal d'innovar.

Pensem, doncs, en moltes funcionalitats i moltes eines, analitzem el mercat per veure fins a on arriben els productes actuals i finalment dibuixem sobre paper els mòduls que compondrien el nostre projecte ideal.

1.2. *Estructura de la memòria del projecte*

Hem organitzat la memòria en els apartats més importants en quant el desenvolupament d'un projecte d'aquest tipus requereix.

Per començar, el present capítol ens serveix a mode d'introducció i resum del que parlarem en més detall en els capítols següents al llarg del document. Ens serveix per donar unes pinzellades de què tracta el projecte, és a dir, quin problema hem detectat en el món laboral i com el volem resoldre.

En el segon capítol presentem un complet estudi de viabilitat on hem analitzat en detall tots els aspectes que intervenen en la construcció del nostre prototip per poder decidir si era factible portar-lo a terme o no. Dins d'aquests aspectes trobem la relació d'objectius de projecte i requisits, anàlisi de la situació actual, és a dir, quin problema volem resoldre i què s'està fent actualment per solucionar-lo, què proposem nosaltres, quines alternatives semblants ja es troben disponibles al mercat, anàlisi de riscos, costos, planificació, etc.

A la tercera secció és la més purament tècnica on es parla de com s'ha plantejat i s'ha portat a terme la solució. És a dir, hi trobem l'anàlisi i el disseny dels mòduls a nivell teòric així com la seva implementació final tot englobat en un mateix apartat per tal de fer un pont entre disseny i implementació i mostrar els seus vincles i diferències.

El tema quatre és dedicat a un resum de la fase de proves a les quals hem sotmès el nostre prototip.

La secció cinc tracta sobre les conclusions a nivell tècnic respecte el producte obtingut partint de la idea inicial i quines futures línies de treball se li podrien aplicar, així com una valoració a nivell personal sobre l'experiència.

Els últims temes són dedicats a glossari, bibliografia i agraïments.

2. Estudi de viabilitat

2.1. *Introducció*

La intenció d'aquest apartat és la de plasmar un estudi de mercat en relació als productes actuals relacionats amb el que es vol realitzar així com de la seva possible acceptació per part dels potencials clients.

Abans de tirar endavant el projecte, s'han d'avaluar riscos i calcular els beneficis esperats per calibrar si l'esforç financer té expectatives de futur.

El punt clau és que amb el nostre prototip volem proporcionar al mercat una eina capaç de respondre a una necessitat real.

2.2. *Tipologia i paraules clau.*

2.2.1. Tipologia:

La tipologia d'aquest projecte té dues vessants:

- a. Desenvolupament software ja que el prototip esperat és un aplicació web.
- b. També de recerca en quan es requereix una investigació algorítmica i de llenguatges formals per a dissenyar una gramàtica generadora de llenguatge capaç de validar les consultes de l'usuari en Llenguatge Natural.

2.2.2. Paraules Clau:

Llenguatge Natural, SQL, llenguatges formals, Ruby, Ruby on Rails, Bases de Dades, Web.

2.3. Descripció.

Partim d'un problema real dins del món laboral.

Es tracta de la necessitat de coneixement sobre el llenguatge estàndard SQL i sistema de base de dades necessari per fer ús de la informació estructurada en entorns de base de dades. Aquesta manera d'organitzar la informació està molt estesa i s'usa en la majoria dels sectors comercials. Inclús els més petits negocis tals com per exemple una floristeria o una cafeteria tenen un sistema de base de dades per, com a mínim, comptabilitzar les vendes per tal de precisar més les futures comandes als proveïdors.

En el nostre cas, presentem un entorn de treball de consultoria informàtica on es treballa amb els conceptes de previsió de vendes, aprovisionament, logística i en definitiva en l'oferta i la demanda de mercats a gran escala.

Hi treballen diferents perfils d'empleat en funció de les seves tasques dins de l'equip de treball.

Podem dividir-los bàsicament en dos grups:

- a. Un perfil funcional amb coneixements econòmics, de màrqueting, analistes de vendes, empresarials... que es dediquen a l'estudi de mercat per tal de generar previsions de vendes, que és el servei que se li ven al client.
- b. Un altre perfil de caire tècnic dedicat a ordenar, actualitzar, reportar, mantenir i administrar les dades de vendes del client contingudes a les bases de dades del projecte. Així com subministrar les solucions i procediments necessaris per tal d'assegurar una eina de treball per als companys de perfil funcional.

Aleshores doncs el problema o mancança en aquest sistema de treball és la falta d'un vincle directe entre el treballador de perfil 'a' i la possibilitat d'extreure d'informació de la bases de dades.

Ja que s'entén que treballadors de tipus b no són subordinats dels de perfil a. És a dir que dins de les seves tasques no es contempla l'extracció de dades de les bases de dades per a la elaboració d'anàlisis de mercat. A part de que tampoc compten amb els coneixements

adequats per tal de comprendre els requeriments funcionals dels anàlisis a fer i en conseqüència de quines són les dades necessàries en cada cas ni un cop obtinguda la informació final són els indicats per raonar el rigor d'aquesta.

Aquí és on entra en joc la nostra idea. Proposem crear un prototip capaç de crear un enllaç directe entre els analistes de vendes i la informació continguda a les bases de dades.

2.4. Objectius del projecte.

La idea és que mitjançant el nostre sistema, el treballador funcional pugui obtenir les dades necessàries usant preguntes en un llenguatge natural que ja coneix.

En el nostre cas el llenguatge natural escollit serà un subconjunt de la llengua Anglesa generat per una gramàtica dissenyada per tal de que contempli les paraules necessàries per a construir el lèxic de les consultes. La intenció és que aquest llenguatge sigui el més proper possible al que usem els humans al parlar la llengua germànica.

D'aquesta manera el treballador podria obtenir les seves fonts d'informació de manera autònoma sense la necessitat d'aprendre un llenguatge de programació com és SQL ni haver de familiaritzar-se amb l'entorn de funcionament de les bases de dades ni tampoc haver de prendre temps als seus companys carregant-los de tasques que no formen part dels seus deures habituals.

A més a més es vol que el sistema sigui integrat dins d'un entorn web que requereixi de la mínima participació per part dels usuaris per tal de que sigui el més senzill possible, que no requereixi instal·lacions, que es pugui consultar des de qualsevol punt a través d'Internet i que no carregui amb més tasques als usuaris perquè el que volem és, en definitiva, agilitzar la seva feina.

Hem triat l'anglès per varies raons. Entre elles perquè d'aquesta manera el prototip podria tenir més sortida al mercat, perquè la llengua oficial de l'empresa Accenture SL és l'anglès, perquè resulta més factible de validar l'anglès com a llenguatge natural que el català pels caràcters que conté, perquè és més coherent fer-ho així ja que la majoria d'entorns de base de dades són en aquesta llengua i si pretenem construir un sistema innovador ha de tenir perspectives de futur i no tancar-se portes. Per aquest motiu, l'aplicació web serà també en anglès així com el manual d'usuari.

2.4.1. Llistat d'objectius.

1. Consultar i obtenir informació, de manera accessible per a l'usuari no expert, d'una base de dades usant un subconjunt del llenguatge natural de la Llengua Anglesa.
2. Gestor web de tiquets per a la monotorització automàtica de l'estat de les consultes.
3. Sistema configurable i amb capacitat d'actualització per tal d'interpretar un subconjunt més ampli de la llengua en el futur.
4. Gestió de la informació de l'estat de les consultes i de l'entorn via correu electrònic.
5. Compte d'administrador pel manteniment de la BD i el sistema.
6. Encapsulament dels resultats en fitxers en format .csv
7. Subministrar serveis web per a poder reutilitzar les funcions del nostre sistema en programari de tercers.
8. Addicionalment dissenyar una aplicació portable executable en local usant les crides a web services.
9. Eina d'autocompletat de camps i/o noms de taules dins de la interfície d'usuari.
10. Manuals d'usuari i administrador.

2.4.2. Catalogació dels objectius

Objectiu #	Crític	Prioritari	Secundari
1	X		
2	X		
3		X	
4		X	
5		X	
6			X
7			X
8			X
9			X
10			X

Taula 1: Taula de catalogació d'objectius.

2.5. Parts interessades.

2.5.1. Stakeholders

Nom	Correu electrònic	Càrrec	Responsabilitat	Cost
Raimon Izard Serrano (RIS)	raimon.izard@gmail.com	Analista Programador.	- Anàlisis de requeriments. - Desenvolupament del projecte. - ...	20€/hora
Jordi Duran Cals (JDC)	jduran@deic.uab.cat	Director de projecte	- Supervisar el bon desenvolupament del projecte així com la seva viabilitat.	40€/hora
Empreses del sector de la consultoria informàtica o qualsevol que faci ús d'entorns dotats de BD amb treballadors no especialitzats.	-	En condició de futuribles clients i/o compradors de l'eina que volem fer realitat.		

Taula 2: Stakeholders que intervenen.

2.5.2. Perfils d'usuari

Tipus	Perfil	Responsabilitat
Equip del projecte.	Administrador del sistema.	Gestionar incidències, millores al sistema, suport.
Eng. Tècnic Informàtic.	Administrador de Base de Dades.	Gestionar incidències, mantenir el sistema en funcionament, introduir noves dades...
Màrqueting, economista, analista de vendes...	Usuari no expert.	Consultar informació a la base de dades usant el sistema en llenguatge natural.

Taula 3: Relació dels diferents perfils d'usuari.

2.5.3. Equip de projecte

Nom	Correu electrònic	Càrrec	Responsabilitat	Cost
Raimon Izard Serrano (RIS)	raimon.izard@gmail.com	Analista Programador. Programador. Investigador. Tècnic de proves.	- Anàlisis de requeriments. - Desenvolupament del projecte. - ...	20€/h
Jordi Duran Cals (JDC)	jduran@deic.uab.cat	Director de projecte	- Supervisar el bon desenvolupament del projecte així com la seva viabilitat.	40€/h
Jaume Pujol Capdevila (JPC)	Jaume.Pujol@uab.cat	Director de projecte adjunt	- Supervisió i suport en la fase inicial i a l'estudi de viabilitat.	30€/h

Taula 4: Relació de l'equip de projecte.

2.6. *Producte i documentació del projecte.*

El resultat del projecte serà l'obtenció d'un prototip capaç de consultar i extreure informació d'una Base de Dades usant subconjunt del llenguatge natural de la Llengua Anglesa.

L'aposta inicial és fer-ho mitjançant una aplicació web accessible per múltiples usuaris des d'arreu del món.

Tenim projectada l'opció de fer una aplicació executable en local com a addicional.

Documentació prevista pel suport del projecte:

- Estudi de viabilitat.
- Memòria del projecte.
- Manual d'usuari per a l'aplicació.
- API web del software.

Com a accions de comunicació:

- Presentació oral davant del tribunal.
- Publicació d'article en blog del *wordpress*¹.

¹ WordPress és una aplicació de publicació d'articles en format de bloc electrònic via web gratuït i de codi obert.

2.7. Estudi de la situació actual

Tal i com ens trobem en la fase inicial del desenvolupament d'aquest projecte, no existeix una figura de client i/o comprador pel qual li desenvolupem l'eina feta a mida; tot i així hem fet un estudi de context sobre la consultoria informàtica *Accenture SL* per tal de donar un exemple empíric del problema que volem resoldre dins del món laboral.

Accenture SL es cataloga com a: *“Una empresa global de consultoria de gestió, serveis tecnològics i outsourcing”*.

Així doncs marca el seu objectiu com: *“Accenture col·labora amb els seus clients per ajudar-los a convertir les seves organitzacions en negocis i administracions públiques d'alt rendiment”*².

Si entrem el valors de facturació i empleats podem mencionar que actualment compta amb més de 177.000 empleats repartits en més de 120 països dels cinc continents i darrer exercici del 2009 es va tancar amb una facturació de 21.580 milions de dòlars.

El que a continuació trobem exposat, no és una descripció única del model de negoci de l'empresa en qüestió, si no que és una situació generalitzada dins de les empreses del sector.

Així doncs, actualment dins d'aquest entorn laboral el buit que existeix per la manca del nostre producte s'està suplint a través de dues vies complementàries:

- a. Els empleats l'àrea de coneixement dels quals no inclou l'ús, comunicació i funcionament d'entorns amb Base de Dades, s'esforcen i dediquen un temps en assolir aquest coneixement a la vegada que l'empresa inverteix en cursos de formació interns.
- b. Es realitzen estudis i documents funcionals per a cada una de les consultes a fer a la BD i aquesta informació es tramet al treballador amb els coneixements adients que s'ocupa de dur a terme la tasca redactant el document tècnic corresponent i executant la consulta SQL a través del gestor de BD. Evidentment amb la conseqüent pèrdua de temps degut al traspàs de la idea i dels requeriments de la tasca a realitzar.

² Informacions extretes de la web oficial de la franquícia espanyola de l'empresa:
http://www.accenture.com/Countries/Spain/About_Accenture

Així com el cost econòmic afegit que ha d'assumir l'empresa degut a la necessitat d'ocupar dos treballadors per a fer la mateixa tasca que podria ser assolida per a un de sol.

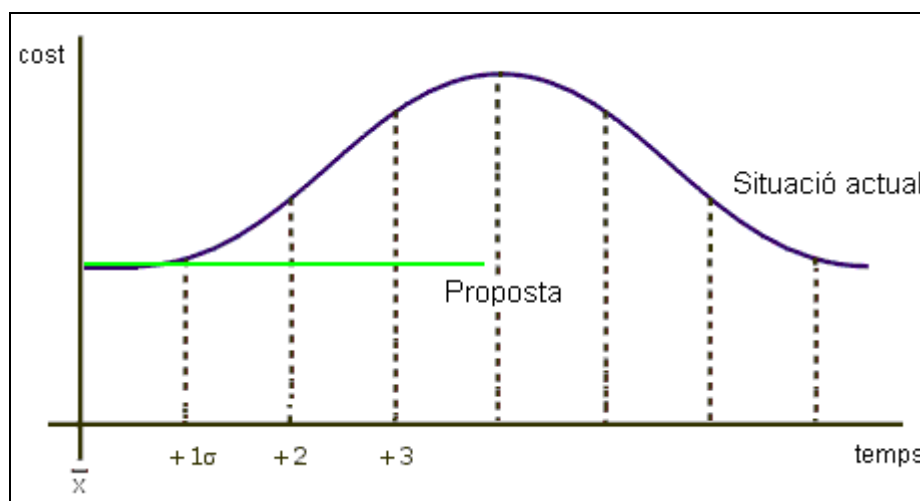
Anant al detall, el tipus de programari usat per a formular explícitament les consultes a la base de dades acostuma a ser un editor de text i un administrador de base de dades. Però tot el software que s'està usant a nivell professional és en llenguatge SQL.

Posant noms de programari podem citar:

- SQLDEVELOPER de Oracle que és un gestor de base de dades freeware però només suporta motors Oracle com a clients de base de dades. Aquest programari ens permet accedir a la informació de les BBDD però només usant SQL.
- TOAD de Quest Software és també un gestor de base de dades (*només freeware per alguns dels sistemes de BD*). Té les mateixes limitacions que l'anterior en quant a l'ús del llenguatge.

Ja que d'editors de text n'hi ha molts dins del mercat i amb tot tipus de llicència d'usuari no els hem comentat.

A la il·lustració següent (Il·lustració 1) podem observar una gràfica virtual aproximada on es plasma l'ús dels recursos en forma del cost associat i el temps necessari per a realitzar una consulta d'informació de la BD amb la situació actual i el previst amb la solució proposada.



Il·lustració 1: Gràfica representativa de l'estudi del cost/temps treballant amb les eines actuals o amb la nostra proposta.

Com podem veure, la gràfica de la funció que representa el cost i el temps necessari per completar la tasca tal i com s'està portant a terme actualment respon a una Gaussiana on

l'increment correspon la dedicació dels dos empleats alhora al llarg del temps i els extrems representa quan només un d'ells hi està treballant. A l'inici seria el treballador funcional, durant el procés hi treballarien els dos fins al punt àlgid amb dedicació màxima per ambdues parts i al final per part del treballador funcional a soles un altre vegada un cop adquireix les dades que volia a l'inici.

En canvi tal i com mostra la línia horitzontal, l'ús dels recursos és molt inferior així com es redueix aproximadament a la meitat el temps necessari.

El resultat d'executar aquest tipus d'activitat varies vegades al llarg de la jornada laboral té un cost acumulat fortament computable ja que s'ha de tenir en compte que la dedicació progressiva dels dos empleats fa que cada un d'ells deixi de dedicar temps als seus deures que el rol d'empleat els ocupa.

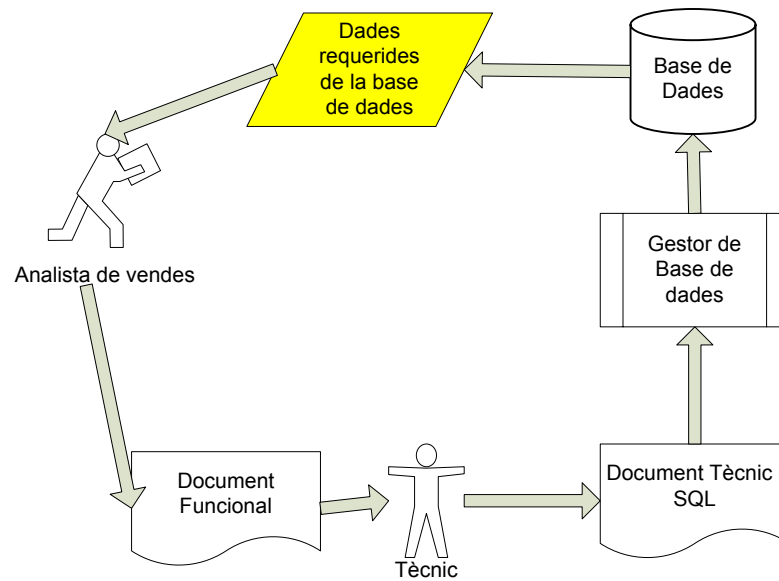
Aleshores el factor temps suma dins del pressupost econòmic, perquè deixem de facturar en altres tasques.

2.8. Context.

	Situació Actual	Solució Proposada
Sistema Operatiu	Windows 2003 Server	GNU/Linux Ubuntu 9.10
Motor de servidor web	IIS	Apache
Motor de Base de Dades	Oracle	MySQL
Eina d'accés a Base de Dades	TOAD, SQLDeveloper	Human_SQL, PhpMyAdmin, Emma
Llenguatge utilitzat	SQL	Subgrup de la llengua anglesa
Compatibilitat	Molt limitada	Infinita
Temps d'aprenentatge requerit per a usar-ho a nivell entremig	> 2 mesos	1 dia
Cost total del sistema	El cost resulta difícil de calcular ja que hi intervenen multitud de variables: nombre de treballadors dins de l'empresa, perfils d'empleat, coneixements sobre el sistema, temps disponible, índex de consultes diàries, complexitat de les mateixes, cost de les llicències del programari que s'ha d'adquirir per a configurar l'entorn...	Cost del projecte dividit pel nombre de llicències previstes de vendre.

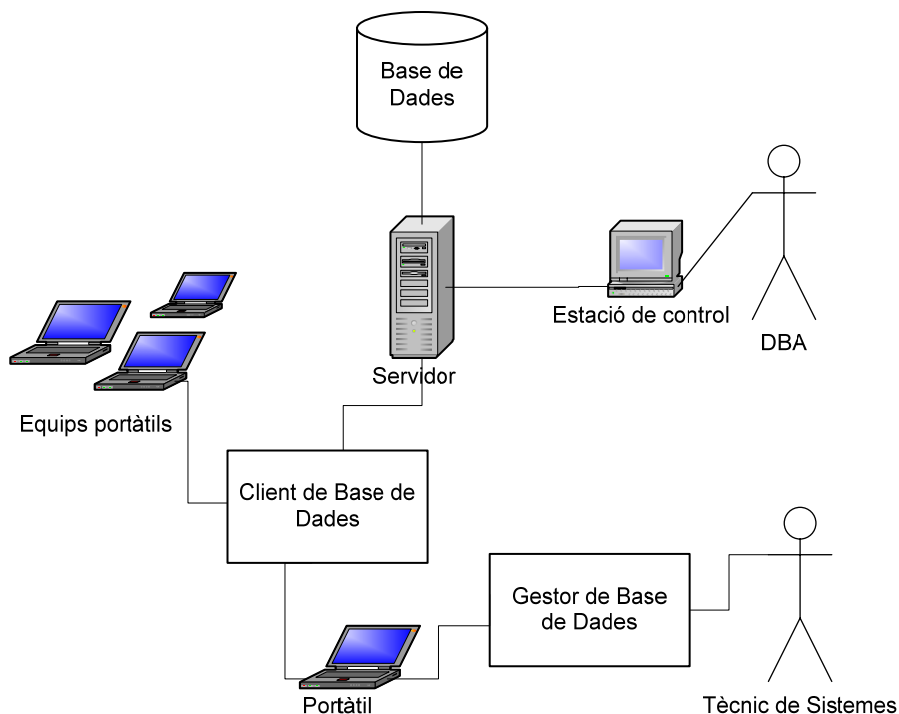
Taula 5: Resum de la situació actual envers la proposta.

2.9. Lògica del sistema.



Il·lustració 2: Graf esquemàtic il·lustrant el flux de treball dins de la situació actual.

2.10. Descripció física.



Il·lustració 3: Esquema del maquinari disponible dins del context real.

Actualment el departament de l'empresa disposa del següent maquinari³:

Model	Tipus	Quantitat
HP Compaq nc6220	Portàtil usuari	10
HP Compaq 6710b	Portàtil usuari	20
HP ProLiant sèrie DL	Servidors	8

Taula 6: Maquinari disponible a l'empresa.

³ Tal i com hem comentat a l'inici d'aquest apartat, les dades citades en aquestes descripcions formen part del cas concret de l'empresa estudiada, Accenture SL.

2.11. *Usuaris i/o personal del sistema.*

Tipus	Perfil	Responsabilitat
Departament d'Informàtica	Administrador del sistema.	Gestionar incidències, millores al sistema, còpies de seguretat, proporcionar equips de treball i suport en general.
Eng. Informàtic, Eng. Telecomunicacions.	Analista programador.	Desenvolupament software, configuració a mida pel client.
Màrqueting, economista, analista de vendes...	Analista de vendes.	Generació d'anàlisis de venda, previsió de venda, aprovisionament... usant les dades de la base de dades a través d'eines d'ofimàtica com fulles de càlcul.

Taula 7: Personal del sistema.

Com podem observar aquí falta un rol d'usuari entremig dels Analistes programadors i els Analistes de vendes.

Enlloc d'interposar un rol d'empleat més, per estalviar costos de l'empresa, els analistes de vendes carreguen de feina als analistes programadors amb demandes d'extracció de dades concretes del sistema per tal de poder-los usar en els seus estudis.

En conseqüència, els programadors i els economistes es desvien de les seves tasques produint menys amb més temps degut a les pèrdues de temps d'ambdues parts dedicades a traspasar coneixement per tal de realitzar correctament les extraccions.

Això es deu a que l'analista de vendes sap quines dades vol, però desconeix com estan organitzades dins del sistema de base de dades i tampoc sap com extreure-les.

D'altra banda el programador sap com realitzar extraccions de dades i com estan distribuïdes però desconeix els detalls conceptuals del significat de la informació. Això deriva en que no està prou qualificat per contrastar les dades extretes i validar que aquestes són correctes i les que volia l'analista de vendes.

2.12. Diagnòstic del sistema.

El sistema actual planteja varis problemes i/o inconvenients:

#	Deficiències del sistema actual	Milliores amb la solució proposada
1	Resulta massa costós assolir els coneixements necessaris per a poder-lo usar degudament per als usuaris no experts.	La nostra proposta parteix d'aquest problema, per tant és un dels principals objectius i raó del producte.
2	El seu manteniment és car i feixuc.	El manteniment d'una base de dades sempre pot resultar costós, però és en gran part inevitable tenir un suport expert per al seu manteniment.
3	Pèrdua de temps productiu de dos o més empleats en traspàs d'informació per a portar a terme una única tasca.	Cada empleat podrà dedicar el seu temps a realitzar les seves tasques, si més no, no en perdrà per culpa de les consultes d'informació a la base de dades.
4	Violacions de les polítiques de privacitat.	L'usuari coneixedor del significat de les dades les podrà consultar directament a la base de dades sense intermediaris. D'altra banda, aquest perfil d'usuari, tampoc té accés ni possibilitat de malmetre informació del sistema.
5	Poc estàndard, compatible i costós d'implementar des de zero.	El nostre sistema és tant compatible que no cal adaptar la base existent per a fer-ho funcionar, si no que, el producte s'adapta a qualsevol motlle.

Taula 8: Sistemes actuals envers proposta.

2.13. Normatives i legislació.

Al llarg del desenvolupament del prototip hem de seguir les següents lleis i estàndards:

Concepte legal	Font
Agència Catalana de Protecció de Dades	http://www.apdcat.net/ca/index.php
IEEE Std 830-1998	http://www.unap.cl/metadot/index.pl?id=20061&isa=item&field_name=item_attachment_file&op=download_file
ISO9000	http://www.iso.org/iso/iso_catalogue/management_standards/iso_9000_iso_14000/iso_9000_essentials.htm
ISO20000	http://www.iso.org/iso/catalogue_detail?csnumber=41332
LOPD	https://www.agpd.es/portalweb/canaldocumentacion/common/estandares_resolucion_madrid.pdf
Normativa PFC UAB	http://www.uab.cat/Document/330/254/projectes-normativa20090630.pdf
Oxford Academy School	http://www.theoxfordacademy.org.uk/ http://www.oxfordacademy.org.uk/courses.php
SQL99	International Organization for Standardization, <i>Database Language SQL</i> , ISO/IEC 9075:1992, American National Standards Institute, ANSI X3.135-1992, New York, NY 10036, November 1992.
XML	http://www.w3.org/XML/

Taula 9: Relació de normatives a complir i les seves fonts.

2.14. *Requisits del projecte*

2.14.1. Llistat de requisits funcionals

1. Manteniment i gestió de les dades dels usuaris catalogats per categories segons els seus permisos i/o privilegis. Això correspon a les funcionalitats nou usuari, editar usuari, eliminar usuari.
2. Manteniment i gestió de les consultes a la base de dades a través del sistema. En termes de l'aplicació, afegir consultes en llenguatge natural usant els formularis de l'entorn web, consultar l'estat d'una petició en procés, aturar o eliminar petició.
3. Administrar la base de dades del sistema per part dels administradors.
4. Extracció de resultats en fitxers compatibles amb eines d'ofimàtica dedicades a fulles de càlcul.
5. Gestió de l'estat de les peticions dels usuaris per part del sistema i que aquest mantingui l'usuari informat en tot moment.
6. Carregar peticions massivament al sistema a partir de fitxers plans.
7. Proporcionar llibreries del codi font a tercers usant web services.
8. Autocompletat dinàmic de consultes en el moment de creació/edició accedint a la base de dades i proposant candidats.

2.14.2. Catalogació dels requeriments funcionals

Objectiu #	Essencial	Condicional	Opcional
1	X		
2	X		
3	X		
4	X		
5		X	
6		X	
7			X
8			X

Taula 10: Taula dels requisits funcionals.

2.14.3. Llistat de requisits no funcionals

1. Ha de ser estàndard i adaptable a qualsevol entorn tan de base de dades, com de Sistema Operatiu, hardware dels servidors, xarxes disponibles, connectivitat, etc.
2. El sistema ha de poder suportar varies sessions d'usuari en paral·lel.
3. El codi font del software ha de ser modulats per tal de poder suportar futures ampliacions.
4. Documentar degudament tot el producte per tal de poder-lo traspasar o vendre íntegrament sense necessitat de subministrar serveis de manteniment.
5. Compliment amb les normatives referents al SQL3 en quant al l'ús d'aquest llenguatge de consultes a base de dades per part del sistema.
6. Compliment amb la LOPD referent a les dades dels usuaris així com el contingut de mostra de la base de dades.
7. Compliment amb la normativa vigent sobre la Llengua Anglesa regentada per l'acadèmia oficial de la universitat de Oxford.
8. Compliment amb la normativa sobre l'estàndard XML.

2.14.4. Catalogació dels requeriments no funcionals

Objectiu #	Essencial	Condicional	Opcional
1	X		
2	X		
3	X		
4	X		
5		X	
6		X	
7		X	
8		X	

Taula 11: Requisits no funcionals.

2.15. Restriccions del sistema.

Ja que tant el maquinari com els sistemes operatius usats actualment en aquest sector empresarial és molt variable, el nostre producte ha de ser totalment compatible i adaptable en qualsevol entorn tant hardware com software com hem fixat anteriorment amb el requeriment no funcional 1.

2.16. Catalogació i prioritització dels requisits.

	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RNF1	RNF2	RNF3	RNF4	RNF5	RNF6	RNF7	RNF8
O1		X				X		X		X					X	
O2		X		X	X	X							X			X
O3									X		X	X	X		X	
O4					X											
O5	X	X	X													
O6				X												X
O7							X		X	X	X	X				
O8		X		X		X	X		X	X	X		X			X
O9		X													X	
O10												X				

Taula 12: Relació d'objectius de projecte amb requisits funcionals i no funcionals.

2.17. Alternatives i selecció de la solució

2.17.1. Alternativa 1: SQ-HAL: Natural Language to SQL Translator

Es tracta d'un programa de codi obert per a tractament de consultes a base de dades usant llenguatge natural en Llengua Anglesa.

Web: <http://www.csse.monash.edu.au/hons/projects/2000/Supun.Ruwanpura/>

Una bona opció seria adquirir i adaptar SQ-HAL per a qualsevol base de dades.

És un programa bastant complet capaç d'interpretar un grup reduït de consultes en llenguatge natural però que incorpora opcions interessants però també alguns inconvenients:

Avantatges	Inconvenients
Gratuït, se'ns proporciona el codi font íntegre de forma lícita.	És un programa d'execució en local, és a dir es requereix instal·lar-lo a cada equip que l'hagi d'usar.
Motor analitzador de Llengua Anglesa fet en Perl.	No té interfície web ni possibilitat de consultar informació a la base de dades d'un servidor.
Mòdul d'auto-aprenentatge per part del motor d'anàlisi.	No està mantingut ni tampoc ningú respondrà per ell. No tenim garanties.
Bona interfície gràfica d'usuari.	Resulta lent a l'hora d'interpretar les consultes.
Compatible amb Sistemes Operatius: Windows i Unix-Like.	No podem extreure els resultats en fitxers de sortida.
Compatible amb diferents motors de BD: Multiplex, Proxy i MySQL.	Gramàtica generadora bastant limitada. Com a resultat només interpreta consultes formulades amb una estructura molt rígida.
	S'ha de configurar cada vegada que s'instal·la en un equip nou.
	No pot obtenir resultats més grans del miler de files.
	És monousuari, per tant doncs requereix d'una instal·lació i un equip per cada usuari.
	Per tal d'ampliar el seu motor d'anàlisi ens hauríem de familiaritzar amb el codi subministrat i adaptar-lo, però d'entrada en desconexem els detalls.

Taula 13: Relació d'avantatges i inconvenients entre alternativa 1 i la nostra proposta.

Els costos derivats d'aplicar aquesta alternativa serien nuls en quant a l'adquisició del software però s'encareixen en quant a l'aprenentatge i manteniment. El principal problema doncs, és que aquesta eina no ha estat pensada per a ser distribuïda i integrada dins del món laboral actual, si no, simplement com a mostra i exercici.

2.17.2. Alternativa 2: English Query 2000

És una alternativa desenvolupada per Microsoft Corporation.

Web: <http://technet.microsoft.com/en-us/library/aa224772%28SQL.80%29.aspx>

Aquest sistema és capaç d'interpretar consultes molt complexes en Llengua Anglesa i aquest és el seu principal valor, analitzem els seus pros i contres:

Avantatges	Inconvenients
Analitzador del llenguatge molt potent.	És un programa d'execució bàsicament en local, és a dir es requereix instal·lar-lo a cada equip que l'hagi d'usar. Tot i que es pot configurar per a entorns web.
Fàcil de configurar per a ús local.	S'ha de comprar la llicència de Microsoft SQL Server.
"Garantia de marca".	Només és compatible amb software propietat de Microsoft.
	Només pot accedir a bases de dades d'Oracle i de Microsoft Access.
	Només funciona sota Sistemes Operatius Microsoft Windows de 32 bits.
	Té un motor no configurable ni ampliable per a l'usuari.

Taula 14: Alternativa 2 envers el prototip.

El cost seria bàsicament el que ens marca el preu de la llicència de Microsoft SQL Server 2005 més el cost del temps destinat a l'aprenentatge del software.

En el nostre cas la llicència que necessitaríem seria la "*Workgroup Edition*"⁴ que està destinada a petites empreses i proveeix d'una base de dades sense límit d'usuaris. Això suposaria un cost de 3.900 dollars més uns dos mesos necessaris per a la implantació i familiarització amb l'eina.

⁴ Informació extreta de: <http://www.microsoft.com/sqlserver/2005/en/us/pricing.aspx>.

2.17.3. Solució proposada.

Per una banda SQL-HAL es presenta com a freeware i podria ser millorat considerablement. Però no compleix requisits tan importants com la possibilitat de treballar amb diferents adaptadors de base de dades, no és multi-usuari, no és un sistema distribuït via web, requereix d'instal·lació, etc. i per la seva transcendència resulta molt poc viable que modificant el sistema es pugin arribar a complir.

D'altra banda la proposta de Microsoft requereix el pagament d'una llicència d'ús així com es tracta d'un programari privatiu totalment "blindat" a possibles modificacions i/o adaptacions. Per tant en aquesta segona alternativa, és insuficient o nul·la la possibilitat d'ampliar, millorar i optimitzar el motor d'anàlisi del llenguatge.

Podem observar a continuació una llista amb les millores que planteja la línia de projecte escollida:

- Cost final tenint en compte el preu de les llicències en un dels casos i el cost derivat del manteniment de l'altre.
- Aplicació via web que no presenten cap de les alternatives.
- Possibilitat de desenvolupar una eina *open source* a partir d'estàndards cosa que ens garanteix una continuïtat i acceptació per part del públic.
- No requereix instal·lació prèvia per a començar a usar-lo.
- Múltiples usuaris.
- Planificació automàtica de les peticions.
- Gestió dels canvis d'estat del sistema via correu electrònic.
- Altres.

2.18. Planificació del projecte

- L'eina usada per a la planificació del projecte és una eina *GPL* anomenada *Gantt Planner*⁵.
- El programari candidat a ser usat com a control de projecte serà el conegut controlador de versions *SubVersion*⁶ també de llicència open source.

La data de començament del projecte va quedar fixada al 21 de Setembre del 2009 i la data aproximada de finalització del mateix es planteja cap a finals de Juny del 2010 un cop feta la presentació davant del tribunal que és quan podrem considerar el cicle del projecte com a acabat.

Tot i que segons la primera planificació la part explícita del desenvolupament del projecte conclouria el dia 23 de Gener de 2010 però amb estimació d'aplaçament degut a imprevistos i/o aparició de riscos i a una possible planificació massa optimista.

Basant-nos en l'estimació actual la durada del projecte seria de 19 setmanes, comptem amb tres calendaris de treball diferents, un per cada component de l'equip:

Raimon Izard Serrano (RIS):

1	Laborables	Festius
Dies	Dilluns – Divendres	Dissabte i Diumenge
Hores	16 – 17h 20.00 – 22h	17 – 20h

Taula 15: Calendari de treball de RIS.

Això representen 19 setmanes * 21 hores de dedicació setmanal = 400 hores.

Jordi Duran Cals (JDC):

2	Laborables	Festius
Dies	Dilluns i Diumenge	Dimarts – Dissabte
Hores	16 – 17h	-

Taula 16: Calendari de participació al projecte de JDC.

En aquest cas, 19 setmanes * 2 hores = 38 hores.

⁵ Podeu obtenir més informació sobre el programari a <http://live.gnome.org/Planner/Screenshots>.

⁶ Web d'informació: <http://subversion.tigris.org/>

Jaume Pujol Capdevila (JPC):

3	Laborables	Festius
Dies	Dijous	Resta
Hores	17 – 19h	-

Taula 17: Calendari de l'adjunt JPC.

En aquest recurs tenim un total de 19 setmanes * 2 hores setmanals = 38 hores.

2.19. Recursos del projecte.

2.19.1. Recursos humans

Aquests són els mateixos que els comentats anteriorment quan ens hem referit a l'equip de projecte.

Nom	Correu electrònic	Càrrec	Responsabilitat	Calendari	Cost
Raimon Iazard Serrano (RIS)	raimon.izard@gmail.com	Analista Programador.	- Anàlisis de requeriments. - Desenvolupament del projecte. - Proves. - Documentació. - Implementació. - Manteniment.	1	20€/hora
Jordi Duran Cals (JDC)	jduran@deic.uab.cat	Project Manager	- Supervisar el bon desenvolupament del projecte així com la seva viabilitat i finalització.	2	40€/hora
Jaume Pujol Capdevila (JPC)	Jaume.Pujol@uab.cat	Project Manager adjunt.	- Supervisió i suport en la fase inicial, a l'estudi de viabilitat, documentació i exposició oral.	3	30€/hora

Taula 18: Informació dels actors dins del desenvolupament del projecte.

2.19.2. Recursos materials

Maquinari disponible

Maquinari	Tipus	Característiques	Propietari	Disponibilitat	Cost	Càrrec pel projecte
ASUS M6Va	Portàtil	<u>Processador:</u> Intel Centrino Dothan 2,13GHz <u>Memòria RAM:</u> 1,5GB DDR2 533MHz <u>Disc Dur:</u> 100GB. <u>T. Gràfica:</u> ATI Radeon X700 128MB <u>Connectivitat:</u> WiFi, Bluetooth. <u>SO:</u> Windows XP Pro. SP3 / Ubuntu Linux 8.04 LTS	RIS	Total	1.700€ (Cost al 2005)	0€
HP Pavilion dm3	Portàtil	<u>Processador:</u> Intel U2300 1,2GHz (x2) <u>Memòria RAM:</u> 4GB DDR3 800MHz <u>Disc Dur:</u> 320GB. <u>T. Gràfica:</u> Intel GMA 4500HD <u>Connectivitat:</u> WiFi, 3G. <u>SO:</u> Windows 7, Ubuntu Linux 9.10.	RIS	Total	550€ (Cost al 2009)	0€
Sap-projectux	Servidor	Hardware desconegut. <u>SO:</u> Debian Linux.	UAB-EUIS	Compartit a temps parcial.	4500€ aprox.	0€

Taula 19: Relació i detalls del maquinari disponible.

Programari disponible

Programari	Llicència	Cost pel projecte
Microsoft Windows XP Pro SP3.	Pagament.	0€*
Microsoft Windows Vista Basic.	Pagament.	0€*
Microsoft Office 2003 Pro amb Visio i Project	Pagament.	0€*
Ubuntu Linux 8.04 LTS	GNU Freeware	0€
Ubuntu Linux 9.10	GNU Freeware	0€
Open Office Suite 2.3	GNU Freeware	0€
Planner	GNU Freeware	0€
Umbrello UML-Modeler	GNU Freeware	0€
Emma MySQL manager	GNU Freeware	0€
MySQL server	GNU Freeware	0€
PHPMyAdmin	GNU Freeware	0€
Apache Server	GNU Freeware	0€
Ruby 1.8 on Rails 2.3	GNU Freeware	0€
Subversion	GNU Freeware	0€
Gitt	GNU Freeware	0€

Taula 20: Programari disponible.

** En el nostre cas, ja comptàvem amb el software privatiu esmentat previ al projecte.*

2.20. Tasques del projecte.

Tasca		Subtasca		Relacions / Dependències		Recursos	Durada dins del projecte*
#	Descripció	#	Descripció	R	D		
1	Fixar els objectius del projecte.	1.1	Analitzar la necessitat del mercat actual i avançar-nos al client.			JDC, RIS	2d 1h
		1.2	Estudiar el sector afectat per la necessitat.	1.1	1.1	RIS	6d
		1.3	Buscar solucions dins del mercat actual.	1.1		JDC, RIS	8d 1h
		1.4	Pluja d'idees.			JDC, RIS	1d
		1.5	Plantejar una proposta de producte per solucionar la manca.	1.4	1.3	JDC, RIS	8d
		1.6	Fixar objectius, extrems i límits.		1.5	JDC, RIS	1d 1h
		1.7	Estudiar la possible acceptació del producte per part del client.	1.1 1.2 1.4 1.5	1.5 1.6	JDC, RIS	1d
		1.8	Estudi de viabilitat.			JPC, RIS	13d 1h
2	Fixar la base de recursos del projecte.	2.1	Marcar els suports que necessitem.		1.8	JDC, RIS	1d
		2.2	Anàlisi de sistema Operatiu; (cost, manteniment, compatibilitat...)		2.1	RIS	1d 1h
		2.3	Programari requerit, el seu cost i adquisició.	2.2		RIS	6d
		2.4	Escollir i configurar servidor web.	2.2 2.3		RIS	10d
3	Configurar el sistema de Base de Dades.	3.1	Dissenyar el model relacional, diagrama entitat-relació.	3.1	3.1	RIS	3d
		3.2	Analitzar els diferents motors de BD disponibles i compatibilitats.			RIS	1d
		3.3	Fer realitat el disseny a través de les eines de Ruby dins del motor de BD.		3.2	RIS	6d
		3.4	Accés i manteniment de a la BD a través d'algun gestor de BD.	3.3		RIS	2d 1h
		3.5	Graella de tests de la BD així com estudis d'escalabilitat i muti-usuari.			JDC, RIS	7d
4	Desenvolupar	4.1	Estudiar Framework Rails.			RIS	10d

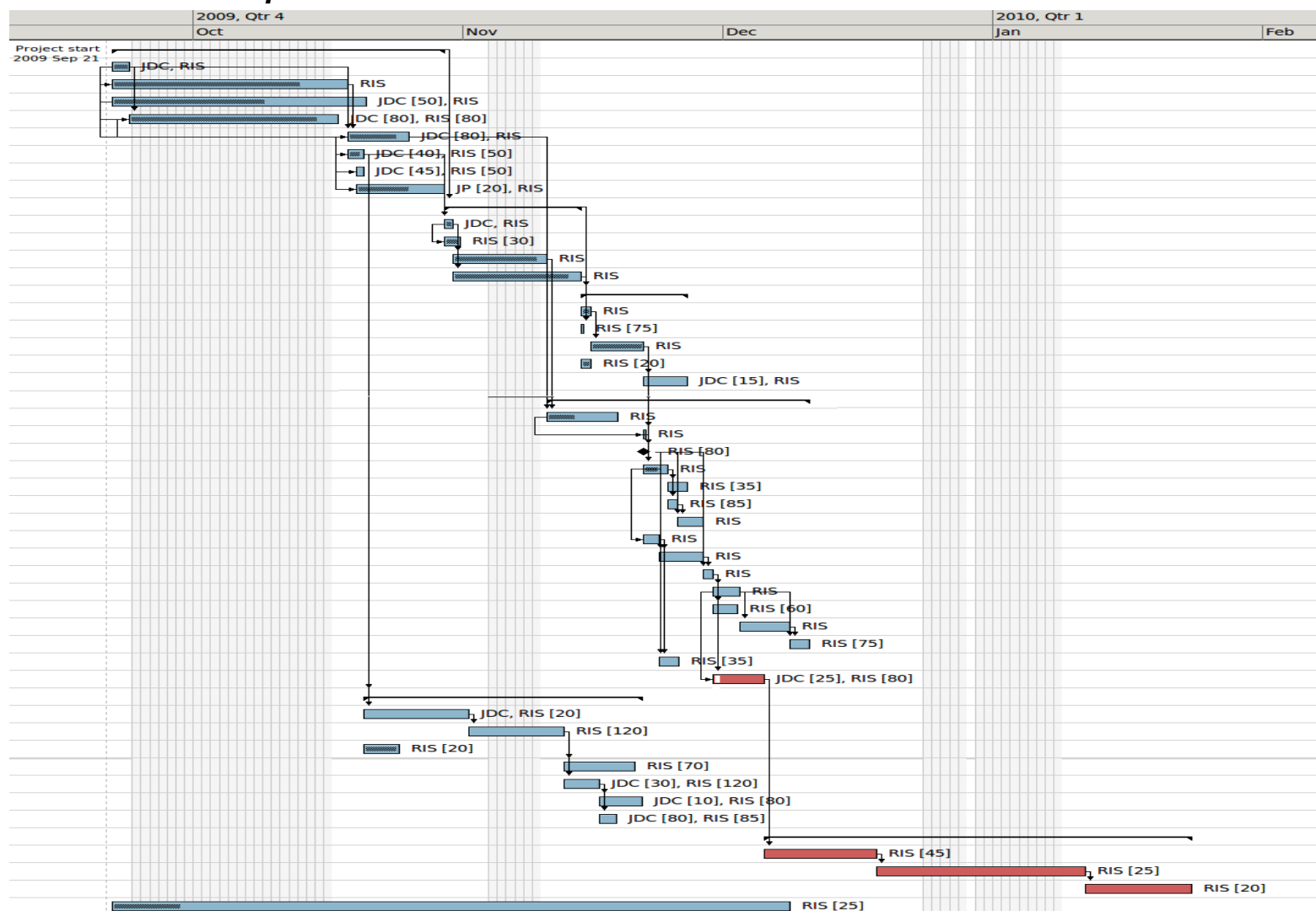
		4.2	Configurar el sistema, dissenyar models de dades.			RIS	1d
		4.3	Gestionar la BD a través de Rails.		4.2	RIS	3d
		4.4	Generar vistes per a cada model.	4.3	4.2	RIS	3d
		4.5	Enllaçar vistes.		4.4	RIS	2d 1h
		4.6	Crear menú principal.			RIS	1d
		4.7	Aplicar estil.			RIS	3d
		4.8	Rols d'usuari.			RIS	3d
		4.9	Implementar seguretat.		4.8	RIS	5d
		4.10	Aplicar funcionalitat per afegir i gestionar consultes en PLN.		4.8	RIS	3d
		4.11	Gestionar fluxos de dades, extracció de dades de la BD i sortides a fitxer.			RIS	3d
		4.12	Afegir autocompletat de paraules camp i noms de taula.		4.10	RIS	2d 1h
		4.13	Implementar gestor de tiquets.		4.11	RIS	7d
		4.14	Configurar enviament de correus electrònics automàtics del sistema a		4.11	RIS	2d 1h
		4.15	Múltiples peticions d'usuaris.			RIS	2d 1h
		4.16	Graella de testos.			JDC, RIS	6d
5	Motor de PLN.	5.1	Dissenyar a nivell teòric la gramàtica i el seu autòmat.			JDC, RIS	13d 1h
		5.2	Implementar la gramàtica en Ruby de forma modular i ampliable.		5.1	RIS	5d 1h
		5.3	Definir un estàndard de dades a tractar (XML, JSON...)	5.2		RIS	5d
		5.4	Tractament dels fitxers amb l'estàndard escollit en solitari.		5.3	RIS	10d
		5.5	Inserir el motor dins del sistema complex de la BD, SiteWeb.		5.2	JDC, RIS	5d 1h
		5.6	Graella de Testos.		5.6	JDC, RIS	4d 1h
		5.7	Fer accessible el motor a peticions externes a través de WebServices.		5.5	JDC, RIS	2d
6	Desenvolupament d'aplicació executable en local.	6.1	Crear un executable que es proveeixi de les llibreries subministrades pel WebService del sistema.		4.18	RIS	15d 1h
		6.2	Graella de testos.		6.1	RIS	12d
		6.3	Manual d'usuari.		6.1	RIS	15d

7	Documentació de disseny, annexos i altres.				1.5	RIS	60d

Taula 21: Taula de les tasques del projecte planificades al Novembre de 2009.

**En aquesta taula hem aplicat els temps de durada de tasca segons la planificació. És a dir, tenint en compte les dependències entre tasques i les que es poden executar en paral·lel. Aleshores els dies i es hores es basen en els calendaris de treball de cada un dels actors, no en dies naturals.*

2.21. Planificació temporal.



Il·lustració 4: Diagrama de Gantt de la planificació del projecte feta al Novembre.

2.22. Avaluació de riscos

2.22.1. Llista de riscos.

#	Risc	Origen	Descripció	Efecte
1	No poder dissenyar un autòmat capaç de mantenir gramàtica generadora del subconjunt de la Llengua Anglesa.	Disseny del motor d'anàlisi de les consultes en llenguatge natural.	Es podria donar el cas de que no poguéssim implementar un bon sistema d'anàlisi del llenguatge natural i en conseqüència no poder tractar les consultes en llenguatge natural.	L'efecte seria devastador perquè és un dels dos punts fonamentals del projecte juntament amb l'entorn del sistema.
2	No poder agrupar tots els mòduls del sistema perquè treballin conjuntament.	Modularitat i compatibilitat dels components.	Si no podem ajuntar els mòduls no tenim producte final.	Els efectes serien crítics perquè és el segon dels punts forts en què es basa la qualitat del nostre producte. Tot i que en aquest cas els riscos són mínims.
3	No poder cobrir els costos del desenvolupament del projecte i que aquest no es pugui acabar.	Econòmic.	Falta de pressupost per poder completar totes les etapes del projecte degut a mala previsió o algun dels riscos.	Si no podem acabar el projecte no tenim producte i per tant només obtenim pèrdues.
4	No poder acabar el projecte dins del termini de temps establert.	Temps disponible pel desenvolupament.	Endarrerir-nos en alguna de les tasques críiques. Algun dels riscos es manifesta. Mala previsió i ens sorgeix algun imprevist.	Si ens endarrerim però almenys acabem el projecte fora de plaç obtindrem una penalització però tindrem un producte per intentar cobrir costos.
5	Accident fatal amb l'equip informàtic amb el que comptem per desenvolupar el projecte.	Equip informàtic.	Pèrdua de totes les dades del projecte tals com el codi font, com les dades de la Base de Dades i la documentació.	Seria fatal perquè hauríem de començar un altre cop de nou.

Taula 22: Relació de riscos.

2.22.2. Catalogació de riscos.

Risc #	Probabilitat [0..10]	Impacte [0..10]	Total [0..10]
1	5	9	7
2	3	9	6
3	1	10	5,5
4	4	7	5,5
5	2	8	5

Taula 23: Catalogació dels riscos.

2.23. Pla de contingència.

Risc #	Solució	Conseqüències
1	Buscar algun analitzador de llenguatge dins del mercat que s'adapti als nostres requeriments i al nostre sistema.	Costos econòmics afegits pel fet d'adquirir un analitzador de text extern. Costos derivats de superar el temps previst per a la tasca crítica de dissenyar la gramàtica generadora.
2	Renunciar a alguna de les funcionalitats i/o objectius del projecte tals com l'aplicació executable en local, l'extracció de dades, l'enviament de correus electrònics de confirmació, les funcionalitats webserver...	Pèrdua de valor del producte final i no assolir tots els requeriments i objectius del projecte. Pèrdua econòmica.
3	Buscar alternatives més econòmiques. Buscar patrocinadors.	Si tot i adoptar les mesures no resulta suficient, hauríem de renunciar al projecte. Pèrdua del 100% de la inversió principal.
4	Intentar reduir alguna de les tasques crítiques per així ajustar la durada final del projecte.	Possible renúncia a algun dels objectius per tal de reduir la durada final. Si entreguem fora de temps, tindrem penalització econòmica.
5	Fer còpies de seguretat periòdicament durant el desenvolupament del projecte. Així com usar un programa de control de versions del software.	Impacte moderat dins del projecte en funció de la magnitud del desastre. En tot cas fer còpies és una bona solució i no comporta una sobrecàrrega important.

Taula 24: Resultat d'aplicar les solucions als riscos comentats anteriorment.

2.24. Pressupost

2.24.1. Estimació cost de personal.

Tal i com hem calculat anteriorment a l'apartat 5 quan hem parlat de la planificació i dels recursos humans disponibles:

	Hores	Preu/Hora	Cost total
Analista programador / Tècnic de proves.	401,35 h*	20 €	8027€
Cap de projecte.	38 h	40 €	1520 €
Cap de projecte adjunt.	38 h	30 €	1140 €

Taula 25: Justificació costos de personal.

**S'ha calculat en funció del cost de les tasques marcat per la planificació, per això hi tenim una desviació de 1,35h necessàries per acabar el projecte dins del pla establert.*

La suma total derivada dels recursos és de 10687 €.

2.24.2. Estimació cost dels recursos.

Tal i com s'ha detallat a l'apartat 5 quan hem parlat dels recursos de maquinari i de programari dels que comptem per al projecte, aquests no apliquen un recàrrec addicional al pressupost.

Tot i així en podem calcular el cost que suposaria si se n'hagués de fer càrrec el projecte.

Concepte	Cost amortització	Cost unitari	Període amortització	Període utilització
ASUS M6Va	170€	1700€	42 m	4,2 m
HP Pavilion dm3	72.62€	415€	24 m	4,2 m
Sab-Projectux	315€	4500€	60 m	4,2 m
Llicència Microsoft Windows XP SP3.	14€	120€	36 m	4,2 m
Llicència Microsoft Office amb MSProject i Visio.	14€	120€	36 m	4,2 m

Taula 26: Desgloçament costos de maquinari.

Això representa un total de 585.62 € de cost de recursos materials.

2.24.3. Estimació cost de les activitats.

Taula d'estimació de costos per activitat realitzada a Novembre de 2009.

WBS	Tasca	Inici	Final	Duració *	Prioritat	Complet	Cost (€)
1	Fixar els objectius del projecte.	Sep 21	Oct 29	41d 1h			3067.22
1.1	Analitzar la necessitat del mercat actual i avançar-nos al client.	Sep 21	Sep 23	3d	1	100%	300
1.2	Estudiar el sector afectat per la necessitat.	Sep 21	Oct 18	6d	2	80%	240
1.3	Buscar solucions dins del mercat actual.	Sep 21	Oct 20	9d	3	60%	680
1.4	Pluja d'idees.	Sep 23	Oct 17	1d	2	90%	120
1.5	Plantejar una proposta de producte per solucionar la manca.	Oct 18	Oct 25	9d	1	80%	879.67
1.6	Fixar objectius, extrems i límits.	Oct 18	Oct 20	1d	4	80%	83.78
1.7	Estudiar la possible acceptació del producte per part del client.	Oct 19	Oct 20	1h	4	40%	56
1.8	Estudi de viabilitat.	Oct 19	Oct 29	12d	2	60%	707.78
2	Fixar la base de recursos del projecte.	Oct 29	Nov 14	17d 1h			780
2.1	Marcar els suports que necessitem.	Oct 29	Oct 30	1d	2	90%	120
2.2	Anàlisi de sistema Operatiu.	Oct 29	Oct 31	1h	4	90%	20
2.3	Programari requerit, el seu cost i adquisició.	Oct 30	Nov 10	6d	4	90%	240
2.4	Escollir i configurar servidor web.	Oct 30	Nov 14	10d	2	90%	400
3	Configurar el sistema de Base de Dades.	Nov 14	Des 4	17d 1h			789.78
3.1	Dissenyar el model relacional, diagrama entitat-relació.	Nov 14	Nov 15	3d	2	100%	120
3.2	Analitzar els diferents motors de BD disponibles i compatibilitats.	Nov 14	Nov 14	1d	4	100%	40
3.3	Fer realitat el disseny a través de les eines de Ruby dins del motor de BD.	Nov 15	Nov 21	6d	3	100%	240
3.4	Accés i manteniment de a la BD a través d'algun gestor de BD.	Nov 14	Nov 15	1h	5	100%	20
3.5	Graella de tests de la BD així com estudis d'escalabilitat i multi-usuari.	Nov 28	Des 4	7d	2	0%	369.78
4	Desenvolupar Web Site.	Nov 10	Des 8	52d 1h			2217.31
4.1	Estudiar Framework Rails.	Nov 10	Nov 18	10d	4	40%	400
4.2	Configurar el sistema, dissenyar models de dades.	Nov 21	Nov 21	1d	3	90%	40
4.3	Gestionar la BD a través de Rails.	Nov 21	Nov 24	3d	3	50%	120
4.4	Generar vistes per a cada model.	Nov 21	Nov 24	3d	3	60%	120
4.5	Enllaçar vistes.	Nov 24	Nov 26	1d	5	0%	40.06
4.6	Crear menú principal.	Nov 24	Nov 25	1d	4	0%	40.14

4.7	Aplicar estil.	Nov 25	Nov 28	3d	4	0%	120
4.8	Rols d'usuari.	Nov 21	Nov 23	3d	3	0%	120
4.9	Implementar seguretat.	Nov 23	Nov 28	5d	3	0%	200
4.10	Aplicar funcionalitat per afegir i gestionar consultes en PLN.	Nov 28	Nov 29	3d	2	0%	120
4.11	Gestionar fluxos de dades, extracció de dades de la BD i sortides a fitxer.	Nov 28	Nov 30	3d	4	0%	120
4.12	Afegir autocompletat de paraules camp i noms de taula.	Nov 29	Des 2	1d 1h	5	0%	60
4.13	Implementar gestor de tiquets de consultes.	Nov 30	Dec 6	7d	2	0%	280
4.14	Configurar enviament automàtic de correus electrònics a l'usuari.	Des 6	Des 8	2d	4	0%	80
4.15	Múltiples peticions d'usuaris.	Nov 23	Nov 25	1d	3	0%	40.06
4.16	Graella de testos.	Nov 29	Des 5	5d	3	0%	317.06
5	Motor de PLN.	Oct 20	Des 23	32d			2712.75
5.1	Dissenyar a nivell teòric la gramàtica i el seu autòmat.	Oct 20	Nov 1	4d	2	0%	1284.33
5.2	Implementar la gramàtica en Ruby de forma modular i ampliable.	Des 11	Des 15	7d	2	0%	280
5.3	Definir un estàndard de dades a tractar (XML, JSON...)	Oct 20	Oct 24	1d	5	90%	40
5.4	Tractament dels fitxers amb l'estàndard escollit en solitari.	Des 15	Des 23	7d	3	0%	280
5.5	Inserir el motor dins del sistema complex de la BD, SiteWeb.	Des 15	Des 20	7d	2	0%	425
5.6	Graella de Testos.	Oct 20	Oct 24	4d	3	0%	198.89
5.7	Fer accessible el motor a peticions externes a través de WebServices.	Des 20	Des 21	2d	4	0%	205.53
6	Desenvolupament d'aplicació executable en local.	Des 5	Gen 23	13d			520
6.1	Crear un executable que es proveeixi de les llibreries subministrades pel WebService del sistema.	Des 5	Des 18	7d	5	0%	280
6.2	Graella de testos.	Des 18	Gen 11	3d	5	0%	120
6.3	Manual d'usuari.	Gen 11	Gen 23	3d	6	0%	120
7	Documentació de disseny, annexos i altres.	Sep 21	Des 8	15d	3	10%	600
						TOTAL:	10687,06

Taula 27: Justificació del cost de les activitats.

**La durada de les tasques és temps en valor absolut que s'ha de dedicar a cada tasca. La durada de les tasques no correspon a tal i com es mostren en el diagrama de Gantt ja que allà es solapen tasques que poden ser desenvolupades paral·lelament. Per tant doncs, la durada es final del projecte es redueix per aquest fet.*

2.24.4. Estimació d'altres costos.

Com a costos fora de planificació tindríem:

Concepte	Cost
Traspàs d'informació de l'eina per part de l'equip de projecte al client.	2h = 180€
Impressió i maquetació de la documentació.	40€

Taula 28: Justificació de costos extraordinaris.

Això suma un total de 220 €.

2.24.5. Resum i anàlisi cost benefici.

El cost del projecte tenint en compte tots els conceptes suma un total de:

- Desenvolupament del projecte: 10687 €
- Costos d'amortització de material: 585.63 €
- Altres costos: 220 €

Total: 11492.63 €.

El total és elevat, però poc costós tenint en compte l'envergadura del projecte.

Hem de pensar que l'estalvi incremental de les empreses del sector generat a partir de l'ús de l'eina seria molt considerable només tenint en compte les hores de treball que s'alliberen els tècnics per a dedicar-se a altres tasques i els cursos de formació que deixen de finançar. S'ha de tenir en compte que aquesta eina podria ser definitiva i que per contra sempre hi haurà empleats nous dins de l'empresa que no comptaran amb els coneixements necessaris per usar la informació continguda dins de sistemes de bases de dades que necessitaran ser formats.

Per tant més producció, reducció de costos i en definitiva més facturació.

2.25. Conclusions

A continuació podem valorar la viabilitat del projecte amb la següent taula de beneficis/inconvenients:

Beneficis	Inconvenients
Reducció de costos de personal.	Recel per part de les possibles empreses del sector.
Més producció i per tant més facturació.	Recel per part dels usuaris envers l'eina.
Els entorns de base de dades deixen de ser una eina de treball problemàtica per a ser un simple mitjà per aconseguir dades.	Necessitat mínima d'instal·lació del sistema dins de l'empresa.
Empleats més autònoms i versàtils.	Necessitat de formació mínima dels usuaris.
Més cohesió i comprensió dins dels grups de treball.	Amortització total en 5 anys degut al cost del servidor.
Reducció de durada de les tasques.	No totes les formulacions en llenguatge natural poden ser transformades en SQL ja que el llenguatge natural té ambigüitats.
Entorn web sense necessitat d'instal·lació prèvia als equips dels usuaris.	

Taula 29: Resum de pros i contres de portar a terme el projecte.

Per tant Beneficis + Inconvenients = **Projecte viable**.

3. Disseny Tècnic i implementació del Sistema

3.1. *Model de desenvolupament*

S'ha usat un model de desenvolupament evolutiu semblant al model de prototips. Això és perquè partíem d'uns objectius ideals a molt alt nivell conceptual. Aleshores hem anat obrint camí a la vegada que hem anat muntant el prototip per tal d'anar en la línia dels objectius marcats.

El projecte és ambiciós, compta de moltes parts i toca moltes disciplines, per tant l'èxit és el d'aconseguir un prototip capaç de resoldre en gran part els objectius principals amb el desavantatge relatiu de no obtenir un producte final degut al desenvolupament complex i innovador. Cal tenir en compte que sovint cal desenvolupar primer un prototip que intenti ser capaç de portar a terme una idea innovadora abans de d'arriscar-se a projectar un producte final el qual seria molt més costós en cas de que resultés un projecte no viable.

3.2. *Entorn de desenvolupament*

El llenguatge de programació base escollit és Ruby i el seu framework per desenvolupament web anomenat Rails. Aquest llenguatge de programació és apte per a qualsevol plataforma ja que és interpretat, però s'adapta millor a sistemes Unix-Like durant les etapes de programació. Per aquest motiu ens hem decantar-nos per un entorn GNU/Linux Ubuntu 9.10 i també per reduir costos ja que es tracta d'un sistema operatiu open source.

Sobre el llenguatge de programació *Ruby on Rails*, podem fer ús de la cita introductòria continguda en el llibre "*Ruby on Rails*" de Bruce A. Tate i Curt Hibbs citat a la bibliografia, que diu:

"El fenómeno Ruby on Rails está arrasando en nuestro sector, con una falta de respeto absoluta hacia los lenguajes de programación consolidados, hacia las convenciones tradicionales o el apoyo comercial."

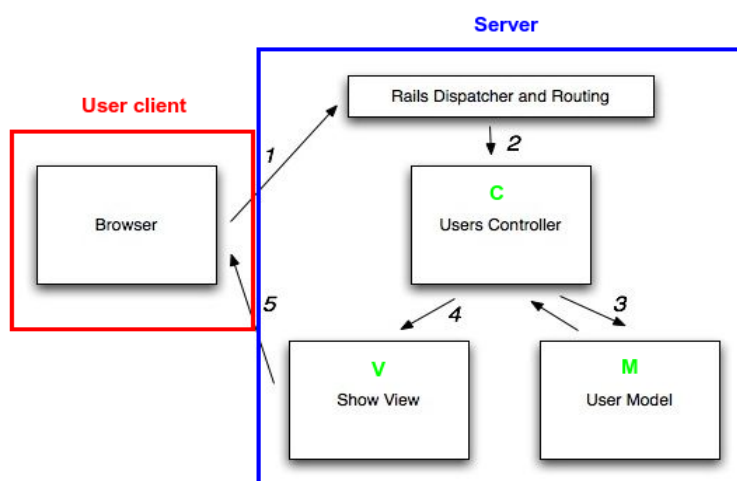
Ruby, amb l'afegit de Rails, és un llenguatge interpretat de 4rta generació amb un paradigma de programació basat en l'orientat a objecte però que va més enllà oferint la filosofia *Model View Controller* orientada a entorns web gràcies a les llibreries que incorpora Rails.

Aquest paradigma ha estat dissenyat per a desenvolupar entorns web amb un model relacional de base de dades per darrera i es basa de tres actors: el controlador, el model i la vista.

El model conté l'estructura d'un tipus de dada abstracte, si fem l'analogia amb el paradigma d'orientat a objecte seria l'equivalent a una classe; que a la vegada correspon a una taula de la base de dades. Les seves instàncies són objectes que cada un d'ells indexen i representen una fila de la taula. En el nostre sistema *'user'* i *'query'* són els nostres models que corresponen a les taules de la base de dades com veurem més endavant. Aquests models es defineixen usant codi de la llibreria Rails.

Les vistes són bàsicament codi xHTML que poden contenir accions, funcions i variables Ruby. S'usen per mostrar d'una manera concreta la informació que contenen els models. Aleshores doncs podem tenir varies vistes per un mateix model segons els perfils d'usuari que tinguem al sistema i de les accions que es puguin aplicar sobre cada model. Per exemple, pel model usuaris tindrem diferents vistes segons si l'acció és la de crear un nou usuari, editar-lo, eliminar-lo o mostrar-lo.

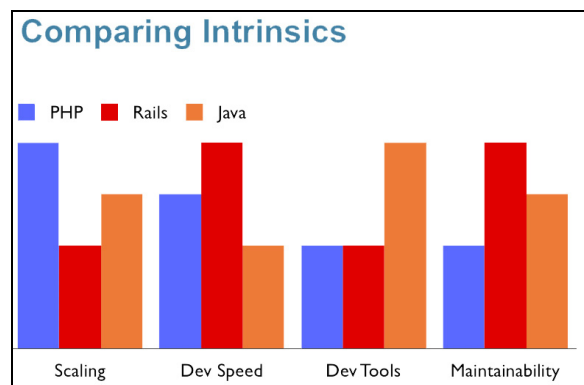
El controlador és bàsicament codi Ruby amb algunes accions de la llibreria Rails i és el responsable d'agafar les dades que ens interessin del model i mostrar-les a través de la vista adient en cada cas. Podem veure a continuació (Il·lustració 5) un esquema del model MVC.



Il·lustració 5: Model de programació web MVC.

Pel nostre prototip necessitàvem disposar d'un entorn web el més ràpid i senzill possible per tal de poder-nos dedicar a la resta de mòduls del sistema ja que la web en si no era l'objectiu final del projecte.

Aquest entorn ens proporciona un desenvolupament molt ràpid i poder obtenir resultats funcionals en un temps breu.



Il·lustració 6: Gràfic de comparació de diferents aspectes entre PHP, Java i Rails.

Com podem observar a la gràfica anterior (Il·lustració 6), extreta de la pàgina web [*PHP vs Java vs Ruby*, <http://www.cmswire.com/cms/industry-news/php-vs-java-vs-ruby-000887.php> - 2010], el framework de Ruby anomenat Rails ofereix un equilibri immillorable entre escalabilitat, velocitat de desenvolupament per part de programador, eines i manteniment. On destaquen la velocitat de desenvolupament i el manteniment que són les dues característiques que més ens interessaven ja que necessitàvem resultats de forma ràpida i un cop fet, tenir la llibertat per modificar i editar el que sigui necessari en el cas de voler afegir noves funcionalitats.

Aleshores doncs s'adapta perfectament al model de desenvolupament de prototips ja que ens permet obtenir un pseudo-producte final ràpidament i poder modificar el que sigui necessari més tard sense perjudicar la resta ni haver d'aplicar grans canvis ja que es tracta d'un llenguatge de programació molt modular.

Podem afegir com a arguments extres per triar aquest llenguatge de programació que està aportant una petita revolució dins del sector de desenvolupament web i que està molt de moda entre els programadors dels Estats Units.

3.3. Disseny del prototip

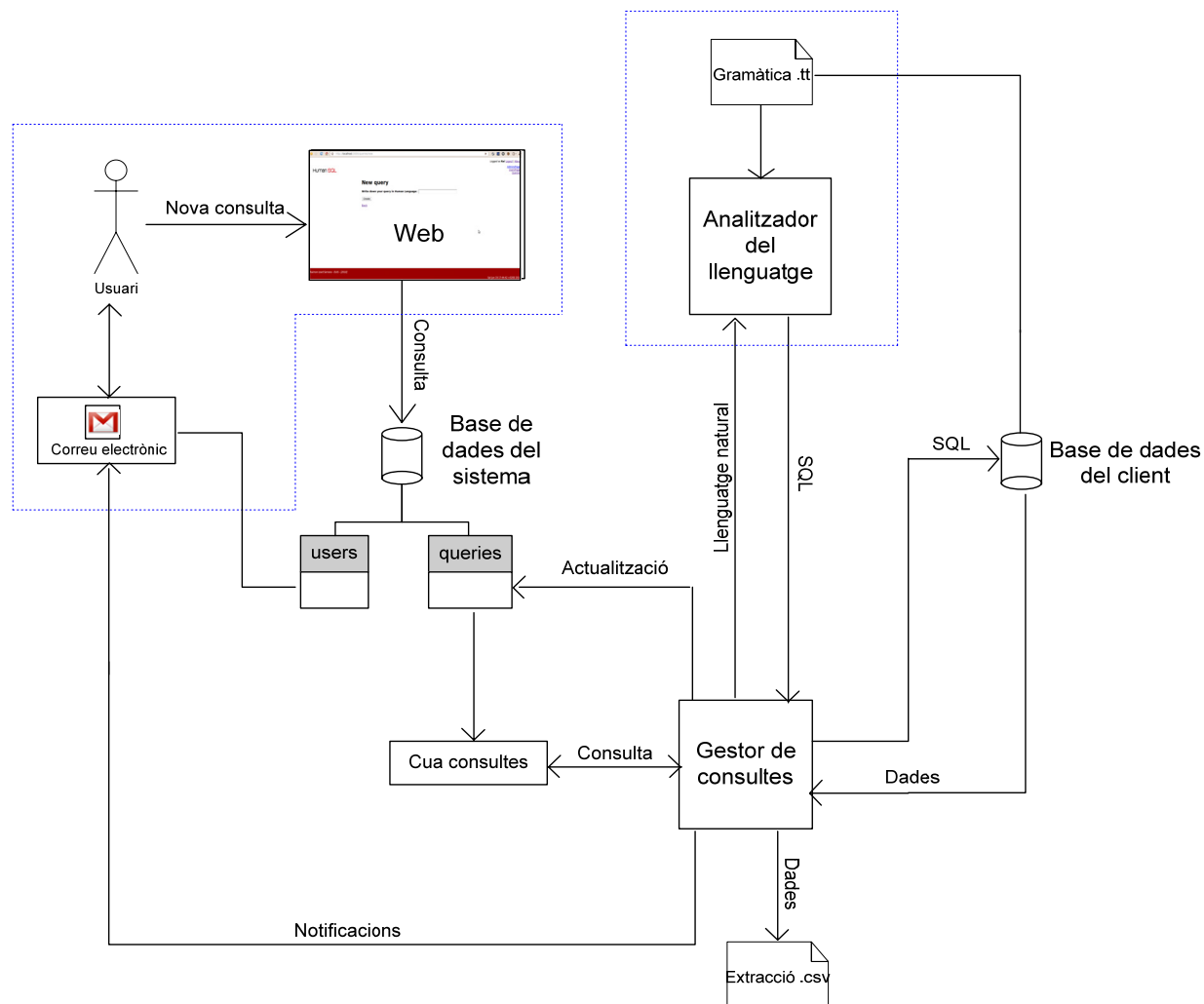
A l'estudi de disseny s'ha identificat el flux principal del sistema que s'inicia en una consulta en llenguatge natural la qual introduïm en un analitzador del llenguatge que fent ús d'una gramàtica BNF és capaç de validar l'entrada i produir una sentència SQL equivalent de sortida.

Tot això englobat en un entorn web que ens serveix d'interfície gràfica per l'usuari final des d'on introdueix noves consultes en llenguatge natural a la base de dades del sistema i un gestor de consultes que controla la cua de peticions pendents al mateix temps que fa funcionar el flux i a més a més extreu la informació de la base de dades del client en un fitxer de sortida usant la sentència SQL produïda per l'analitzador.

El sistema requereix d'una base de dades pròpia per tal de contenir la informació dels usuaris i per tal de guardar l'historial de consultes noves i velles. Les consultes dels usuaris es llancen sobre la informació continguda en una altra base de dades que seria la del client. Aquesta base de dades, no forma part del sistema, si no que estaria allotjada en un altre entorn i des del nostre producte només hi tindrem una connexió de lectura per tal de llançar-hi de manera automàtica les sentències SQL produïdes per l'analitzador de consultes en llenguatge natural i per assegurar-ne la seva integritat, no oferint la possibilitat des del nostre sistema, d'executar sentències SQL que en puguin modificar el seu contingut.

Per tal de que el nostre sistema s'adapti a qualsevol base de dades de qualsevol client i per a què aquest no hagi d'aplicar cap canvi en el seu entorn per incorporar el nostre prototip, hem dissenyat també un mòdul d'anàlisi de la base de dades del client que genera una gramàtica a mida en funció de les taules i columnes.

A la següent imatge (Il·lustració 7) el mapa complet del flux de dades entre tots els mòduls del sistema. Es pot observar que hem emmarcat en una línia discontinua les accions en què intervé l'usuari per una banda i el mòdul d'anàlisi del llenguatge per l'altre ja que és on hi intervenen les llibreries de programació addicionals, com veurem en a l'apartat d'implementació.



Il·lustració 7: Diagrama de tot el sistema.

Mirant el graf, el procés complet del sistema el podem resumir en accions a continuació:

- 1.- Usuari connectat crea una consulta nova a través de la web. A partir d'aquí l'usuari no intervindrà en cap més acció al llarg del procés.
- 2.- La consulta es guarda a la taula '*queries*' de la base de dades del sistema i es posa a la cua de consultes pendents del gestor de consultes.

- 3.- Quan li arriba el torn, el gestor de consultes actualitza l'estat de la consulta, la deriva a l'analitzador de consultes i envia un correu electrònic al propietari informant-lo de que la seva petició està en procés.
- 4.- L'analitzador valida la consulta i produeix una sentència SQL equivalent de sortida que la retorna al gestor de consultes.
- 5.- El gestor de consultes recull la sentència i la llança sobre la base de dades de consulta del client en el cas de que s'hagi processat correctament; si no, s'envia un missatge d'error a l'usuari i aquí acaba el processament de la petició en aquest cas.
- 6.- S'executa la consulta i el gestor en recull els resultats de la base de dades de lectura del client i extreu la informació en un fitxer format .csv. S'actualitza la informació de la consulta a la taula de '*queries*' del sistema aportant la sentència SQL generada i el nom del fitxer de l'extracció de dades. Finalment s'envia un nou correu electrònic al propietari de la petició informant de que la consulta ha esta processada correctament, la sentència SQL produïda respecte la pregunta en llenguatge natural formulada i a on pot trobar l'extracte de dades corresponent.

En el següent apartat a continuació descriurem el disseny i implementació dels mòduls comentats.

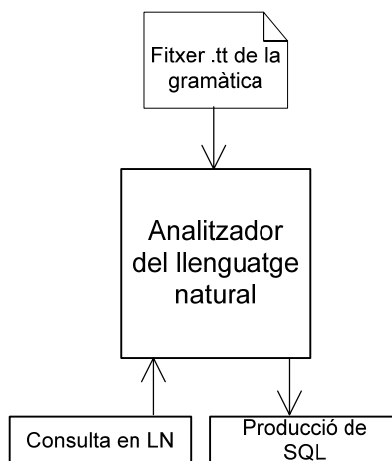
3.4. Disseny i implementació del flux de processament

Aquest flux consta d'una gramàtica generadora d'un subconjunt de la llengua anglesa que valida les consultes d'entrada i a la vegada produeix les sentències SQL equivalents de sortida.

Aquest mòdul consta de dos actors:

- Per una banda tenim el fitxer *.tt*, que és un fitxer de definició de gramàtica BNF⁷ generat a mida de la base de dades de consulta del client en llenguatge *treetop* que és una llibreria de Ruby. Hem usat aquesta eina perquè ens permetia perfecta integració amb Ruby i dissenyar una gramàtica capaç de validar el llenguatge natural.
- D'altra banda tenim les funcionalitats pròpies de la llibreria *treetop* addicional que ens permeten crear un analitzador de llenguatge en funció del fitxer de configuració comentat anteriorment.

A continuació podem veure el flux que segueix el processament de cada una de les consultes dels usuaris (Il·lustració 8):



Il·lustració 8: Diagrama de flux del procés d'anàlisi de les consultes dels usuaris en llenguatge natural.

Com podem veure, el motor analitzador del llenguatge consta de dues entrades:

Per una banda de la gramàtica que defineix el comportament de l'analitzador a través de les normes que defineixen el seu llenguatge formal i les seves produccions.

⁷ Gramàtica BNF segons Wikipedia: http://en.wikipedia.org/wiki/Backus-Naur_form

D'altra banda tenim la petició o paraula a processar per l'analitzador. Parlem de paraula tot i que ens referim a una frase en llenguatge natural, el motiu però és que el *parser*, (analitzador del llenguatge), tracta les paraules de la sentència d'entrada com a lletres del seu llenguatge ja que les normes de la seva gramàtica així ho defineixen. D'aquesta manera ens estalviem de processar lletra per lletra que seria molt més costós i complex.

De sortida retornarà produccions en SQL si l'entrada ha estat validada correctament, si no consumirà l'entrada igualment però generarà un missatge d'error. Per tant doncs, el nostre sistema esdevé determinista ja que consumirà qualsevol paraula d'entrada i en produirà també una sortida en qualsevol cas.

Tot aquest procés de tractament de les consultes pendents és fa *offline*. És a dir que s'executa automàticament independentment de l'estat de l'usuari, si està connectat o no.

S'ha de mencionar que hem usat com a guia la gramàtica en *Perl*⁸ de SQL-HAL (*alternativa 1 de l'estudi de viabilitat*) ja que ofereix una gramàtica de codi obert ben treballada que ens ha permès adaptar-la per construir la nostra base. El funcionament d'aquesta gramàtica és molt similar al de la nostra així com l'arbre de backtracking en profunditat.

Les principals diferències regeixen en el llenguatge de programació usat, en la producció de les sentències SQL equivalents i la manera de fer-les servir. Tot i que la filosofia és la mateixa: validar la consulta en llenguatge natural a la vegada que es va produint la sentència SQL.

3.4.1. Disseny de la gramàtica

Aquest tipus d'analitzadors de gramàtiques anomenats BNF. Funcionen en forma d'arbre de desplegament en profunditat en un paradigma de programació anàleg al *Backtracking*. Es van generant nodes durant l'exploració en profunditat i l'objectiu és el d'arribar a les fulles, nodes terminals, on aleshores podrem recollir les produccions de cada node i tornar a l'origen obtenint la producció final a la sortida.

El sistema comença provant de validar la sentència usant el primer camí, si es queda parat a la meitat perquè ja no pot seguir validant per aquella línia, es descarta aquella branca i es torna enrere fins a l'últim node validat i aleshores s'escull un altre camí.

⁸ Més informació del llenguatge Perl: <http://en.wikipedia.org/wiki/Perl>

Seguirà aquesta estratègia fins a trobar el camí correcte que li permeti consumir totes les paraules d'entrada.

Un exemple senzill adaptat de l'estructura de la nostra gramàtica en *reetop* podria ser el següent:

```
grammar HSQLGrammar
  rule start
    ask table {
      def sql
        sql = ask.out + "FROM " + table
      }
    }

  end

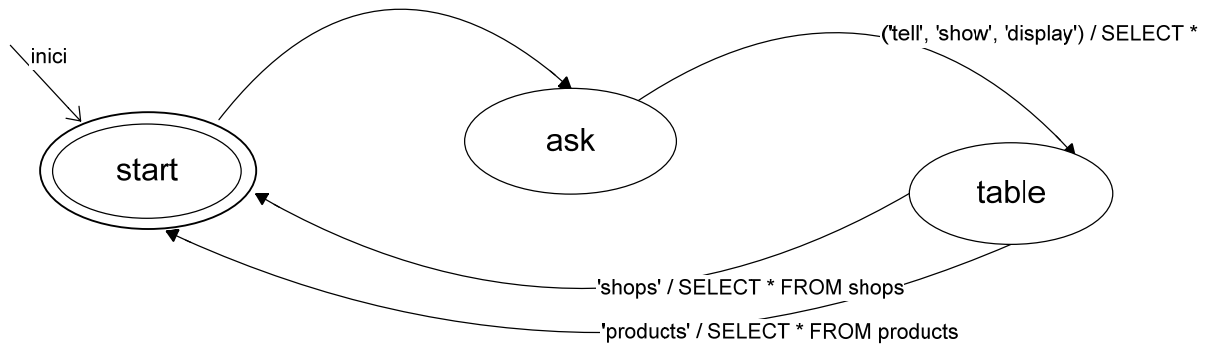
  rule ask
    'tell' / 'show' / 'display' {
      def out
        out = "SELECT * "
      }
    }
  end

  rule table
    'products' / 'shops'
  end

end
```

Aquest fitxer de gramàtica ens validaria les frases d'entrada següents: *'tell products'*, *'display products'*, *'show shops'* i tota la resta de combinacions possibles.

Cal recordar que des del punt de vista de la gramàtica i tal com ja hem comentat a la introducció d'aquest apartat, cada paraula del llenguatge natural és un símbol de la gramàtica. És a dir, no analitzem lletra a lletra, si no que es fa paraula a paraula.



Il·lustració 9: Diagrama d'estats de l'exemple de gramàtica treetop.

Podem veure en el graf anterior com funcionaria la gramàtica d'exemple que hem formulat abans en codi *treetop*. Per exemple, en el cas de tenir una entrada “*tell products*”, la gramàtica validaria la sentència i produiria “*SELECT * FROM products*” com a sortida.

Aquest exemple no és determinista, ja que si no es compleixen les paraules requerides a l'entrada, no s'acabarà de consumir la sentència i no serà validada.

Entrant en detalls, per a dissenyar la gramàtica del nostre sistema, hem dividit les possibles consultes en grups segons el tipus de sentència SQL que s'havia de produir a la sortida.

Aquests grups són: *select*, *average*, *counts*, *sum* i *unknown*.

La idea presa de SQL-HAL, és que agrupem les consultes pel tipus d'estructura i els grups comentats anteriorment esdevenen complementaris en gran part i aconsegueixen definir la major varietat de consultes SQL possibles. Simplement es basa en els diferents operadors de consulta del llenguatge estructurat (*all*, *count*, *average*, *summatory*).

El grup anomenat *unknown* evidentment és en últim lloc ja que quan el sistema no troba cap branca per validar la consulta en llenguatge natural, acaba passant per aquest grup que consumeix qualsevol entrada i sempre genera un missatge d'error de sortida que ens serveix per recollir-lo després i parar el procés. Aleshores doncs parlant en clau de teoria de llenguatges formals, el nostre sistema compta amb una gramàtica generadora de llenguatge que representa un autòmat determinista ja que sempre obtenim una sortida encara que la sentència en llenguatge natural no hagi estat validada.

Dins de cada un dels grups, exceptuant el comentat anteriorment, hi trobem subgrups compartits i d'altres específics que ens permeten aprofundir més i especificar més variants de consultes.

3.5. Disseny de la Base de Dades

El subministrador de base de dades triat és el MySQL i el seu gestor principal via web el *phpMyAdmin*.

Això és perquè hem treballat sota un entorn GNU/Linux Ubuntu 9.10 i resulta molt més còmode, ràpid, lleuger i adaptable treballar amb MySQL. Ja que entre d'altres funcionalitats, ens permet exportar la base de dades completa en fitxers senzills de la mateixa manera que importar-ne una de guardada, consta de versió gratuïta, integració completa amb el sistema operatiu Ubuntu i amb el llenguatge de programació escollit: *Ruby on Rails*.

D'altra banda cal comentar que *Ruby on Rails* permet la connexió directa amb qualsevol dels subministradors de base de dades més comuns tals com DB2, Oracle, Posgress, etc. Cosa que fa que el nostre sistema sigui summament portable a qualsevol entorn ja que s'adapta a la configuració existent.

Pel nostre sistema es requereix una base de dades molt senzilla i òptima de només dues taules. D'altra banda, intervé també una segona base de dades que seria la del client. El nostre sistema llança les consultes sobre aquesta usant una connexió de només lectura per tal de no alterar les dades existents. Cal remarcar doncs, que en el cas de que el nostre prototip esdevingués producte, la base de dades de lectura seria variable en quant a contingut i adaptador. Ja que es tractaria del sistema del client. Això no representaria grans greuges ni modificacions en el nostre sistema, ja que, com ja hem dit, és altament compatible amb la majoria d'adaptadors i estructures de bases de dades relacionals.

3.5.1. Base de dades del sistema

La base de dades del sistema consta de dues taules: *users* i *queries*.

La taula *users* ens serveix per emmagatzemar la informació de tots els usuaris del sistema i per tal de que es puguin connectar i fer servir el producte.

A continuació podem veure la taula usuaris extreta directament de la base de dades usant phpMyAdmin.

Field	Type	Null	Default
id	int(11)	No	
ename	varchar(255)	Yes	NULL
pwd	varchar(255)	Yes	NULL
email	varchar(255)	Yes	NULL
utype	varchar(255)	Yes	NULL
created_at	datetime	Yes	NULL
updated_at	datetime	Yes	NULL
ip	varchar(255)	Yes	NULL
salt	varchar(255)	Yes	NULL

Taula 30: Propietats de la taula users.

Podem comentar l'ús de cada camp:

- **id:** És un índex auto-incremental i és la clau primària de la taula.
- **ename:** És la columna on es guarden els noms d'usuari.
- **pwd:** S'emmagatzema un hash de la contrasenya d'usuari.
- **email:** Guardem el correu electrònic de l'usuari per tal de poder-li enviar missatges quan s'esdevé un canvi en les consultes pendents de l'usuari.
- **utype:** Ens serveix per determinar el grau de permisos de l'usuari. Aquest pot ser 'admin' o 'user' cosa que determina quines taules i quines funcions pot fer cada usuari dins del sistema.
- **created_at:** Es guarda la data d'alta d'usuari.
- **updated_at:** Es guarda la data de l'última modificació de les propietats de l'usuari.
- **ip:** S'emmagatzema immediatament la ip des d'on l'usuari s'ha registrat.
- **salt:** És un cap ocult usat per tal de fer el hash al password d'usuari.

La taula *queries* és la més important del sistema i és la que ens serveix per inserir noves consultes i per veure l'historial de les consultes ja executades.

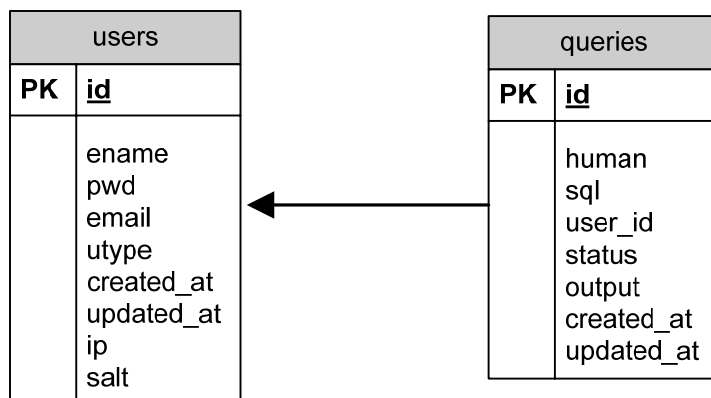
Field	Type	Null	Default
id	int(11)	No	
human	varchar(255)	Yes	NULL
sql	varchar(255)	Yes	NULL
user_id	int(11)	Yes	NULL
status	varchar(255)	Yes	NULL
output	varchar(255)	Yes	NULL
created_at	datetime	Yes	NULL
updated_at	datetime	Yes	NULL

Taula 31: Propietats de la taula *queries*.

Podem comentar l'ús de cada columna:

- **id:** És un índex ocult auto-incremental i és la clau primària de la taula.
- **human:** Aquí s'emmagatzemen les consultes en llenguatge natural introduïdes per l'usuari mitjançant l'interfície web.
- **sql:** Aquest camp l'omple el sistema quan la consulta de l'usuari ha estat "traduïda" del llenguatge natural al SQL.
- **user_id:** Aquest és un camp ocult on es guarda la *id* de l'usuari propietari de la consulta en qüestió.
- **status:** Aquest és un camp informat pel sistema que té una gran utilitat ja que és un agent directe dins de la construcció i manteniment de la cua de consultes pendents a ser processades. Ens informa de l'estat de la consulta i aquest pot ser: pendent, processant, processada o error.
- **output:** Aquí el sistema guarda el nom del fitxer on s'han extret les dades un cop la consulta ha estat processada i el fitxer d'extracció generat.
- **created_at:** Aquest camp del sistema ens serveix per a emmagatzemar la data quan la consulta es va crear.
- **updated_at:** Ens serveix per guardar la data de l'última modificació per a cada entrada de la taula.

Podem veure a continuació el model relacional de les dues taules exposades anteriorment:



Il·lustració 10: Esquema relacional entre les taules del sistema.

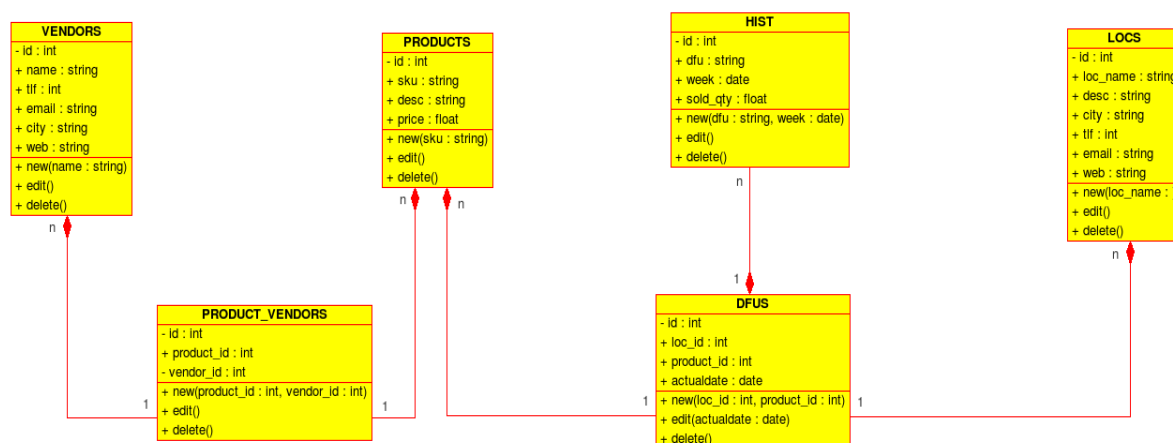
Com es pot observar, cada usuari pot tenir, és a dir ser propietari de 0 o més consultes.

També es dedueix que no podria existir una consulta sense un usuari propietari. Però en el cas de que un usuari propietari de consultes executades i de consultes pendents es donés de baixa del sistema, les processades perdurarien i les pendents serien eliminades. La raó és per tal de conservar l'historial de les processades ja que d'aquesta manera altres usuaris les poden consultar i evitar repetir-les.

Aleshores, el sistema requereix de la instal·lació d'una de base de dades amb el model relacional mostrat anteriorment i existiria una altra base de dades que seria la del client que ja estaria instal·lada prèviament en el seu maquinari.

3.5.2. Base de dades del prototip

Pel prototip del projecte hem usat un possible model relacional de les taules de la base de dades que podria tenir un client del perfil d'*Accenture SL*. D'altra banda, aquesta base de dades sobre la qual es fan les consultes, pot contenir qualsevol informació i qualsevol estructura ja que *Ruby on Rails* s'adapta a qualsevol situació.



II·lustració 11: Model relacional de les taules de mostra usades com a client.

Veiem que és un model clàssic d'empreses consultores on tenim una relació completa entre l'historial de vendes per cada producte a cada botiga per setmanes i la relació del distribuïdor de cada un d'ells.

Cal remarcar que la base de dades del client es farà servir només a mode de lectura des del nostre sistema.

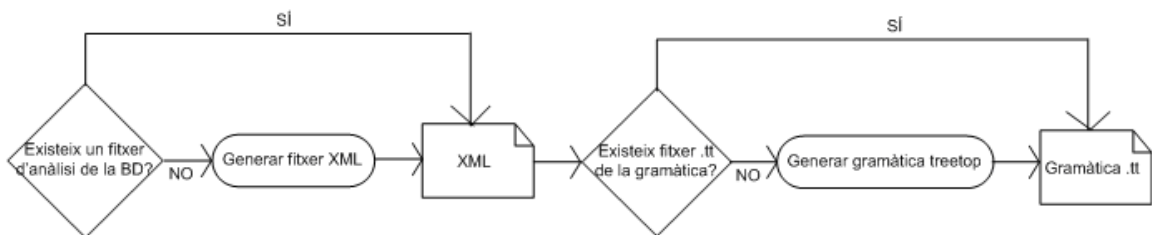
3.5.3. Disseny d'adaptació de la BD del client

Aquest mòdul es dissenya a mode de caixa i fixem els objectius que en volem obtenir:

- Que sigui capaç d'analitzar la base de dades de lectura del client i extreure'n la seva estructura en un fitxer *.xml*.
- Un cop analitzada la base de dades que en generi una gramàtica a mida capaç de validar les consultes en llenguatge natural i produir les sentències SQL equivalents.

D'aquesta manera aconseguim que el nostre sistema sigui capaç d'adaptar-se i de generar una gramàtica a mida segons les característiques de cada sistema de base de dades del client.

A continuació podem veure el diagrama d'estats del mòdul.



Il·lustració 12: Diagrama d'estats del mòdul d'anàlisi i generació de la gramàtica.

Com a norma general, aquest mòdul només actuaria en el moment d'instal·lació i configuració del sistema o quan els fitxers fossin modificats o per expressa voluntat de l'administrador del sistema degut a que per exemple s'han aplicat canvis a la configuració de la base de dades i es vulgui generar una nova gramàtica actualitzada amb el nou estat.

Per a generar el fitxer XML s'usen unes comandes de Ruby per llançar les següents consultes a la base de dades:

```
tables = sql_connection.select_all("Show Tables")

table_names.length.times {|i| columns[i] =
  sql_connection.columns(table_names[i]) }
```

Amb aquestes dues sentències en negreta, vàlides per a qualsevol adaptador de base de dades, obtenim dos vectors un d'ells amb els noms de totes les taules del sistema i

l'altre amb el nom de totes les columnes. S'entén que els noms de taules i columnes del sistema són les relacionades amb la base de dades de lectura del client, no sobre les dades dels nostres usuaris i consultes de *HumanSQL*.

Un cop aconseguida aquesta informació, es genera un fitxer *.xml* amb la següent estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Extract created_at="Wed May 26 19:57:35 +0200 2010" by
Human_SQL-->
<database>
  <name>human_sql_develop</name>
  <adapter>MySQL</adapter>
  <table>
    <num>0</num>
    <name>NomTaula</name>
    <column>
      <num>0</num>
      <name>NomColumna</name>
    </column>
    <column>
      <num>1</num>
      <name> NomColumna </name>
    </column>
    ...
  </table>
  ...
</database>
```

Posteriorment aquest fitxer de configuració ens servirà per generar la gramàtica a mida amb els noms de taules i columnes corresponents.

Per fer-ho, fem servir el següent codi que ens llegeix del fitxer XML i ens escriu a dins de la gramàtica:

```
f_g = File.new("HumanSQL_grammar.tt", "w")
db = File.new("database.xml", "r")
while db.eof == false
  if db.readline.grep(/<table>/) != []
    db.readline
    table_name = db.readline
    x=table_name.index(">")
    y=table_name.index("</")
    table_name = table_name[x+1..y-1]
    rule_tables = rule_tables+"'" +table_name+"'/ "
  end
end

f_g.puts(rule_tables)
```

A continuació podem veure el resultat de l'a funció de les taules dins del fitxer *treetop* de la gramàtica:

```
rule table_name
  'NomTaula1/' NomTaula2/' NomTaula3'...
end
```

D'aquesta manera els noms de les taules poden ser usats per validar les consultes en llenguatge natural.

3.6. Disseny de l'aplicació web

La decisió d'usar un entorn web sorgeix de donar una facilitat extra a l'usuari. És a dir, oferir un sistema integrat i distribuït que no requereix de cap instal·lació ni configuració als equips terminals.

Aquest fet proporciona un sistema molt més àgil i independent ja que tot l'anàlisi i processament del llenguatge de les consultes es fa sense necessitat d'acció per part de l'usuari.

Les eines usades pel desenvolupament web han estat l'editor de text de l'escriptori GNOME de Ubuntu anomenat *gedit* i el terminal o consola del propi sistema operatiu.

Per a fer les proves d'enquadrament i de funcionament s'ha fet servir la versió beta del navegador Google Chrome per Linux ja que es tracta d'un navegador que ofereix l'ample de vista de pàgina web per pantalla més gran de tot el mercat de navegadors i perquè està en constant evolució i creixement. A més a més, consta d'algunes opcions extra que són molt útils per a desenvolupadors web tals com utilitats per esborrar de forma asíncrona les *cookies* i variables de sessió d'una web en concret.

Hem de tenir en compte que estem davant del desenvolupament d'un prototip i per això la seguretat del sistema no ha estat una prioritat ni tan sols un objectiu, ja que se suposa que només es farà servir el sistema per a demostracions i en cap cas directament dins d'un entorn de producció. El disseny i l'estètica de la web tampoc ha estat en cap moment un objectiu o punt clau del prototip ja que ens hem basat en les seves funcionalitats, no en l'aparença.

Tot i així, es va fer un disseny simple, net i funcional on el nom, logotip i imatge del producte és el següent:

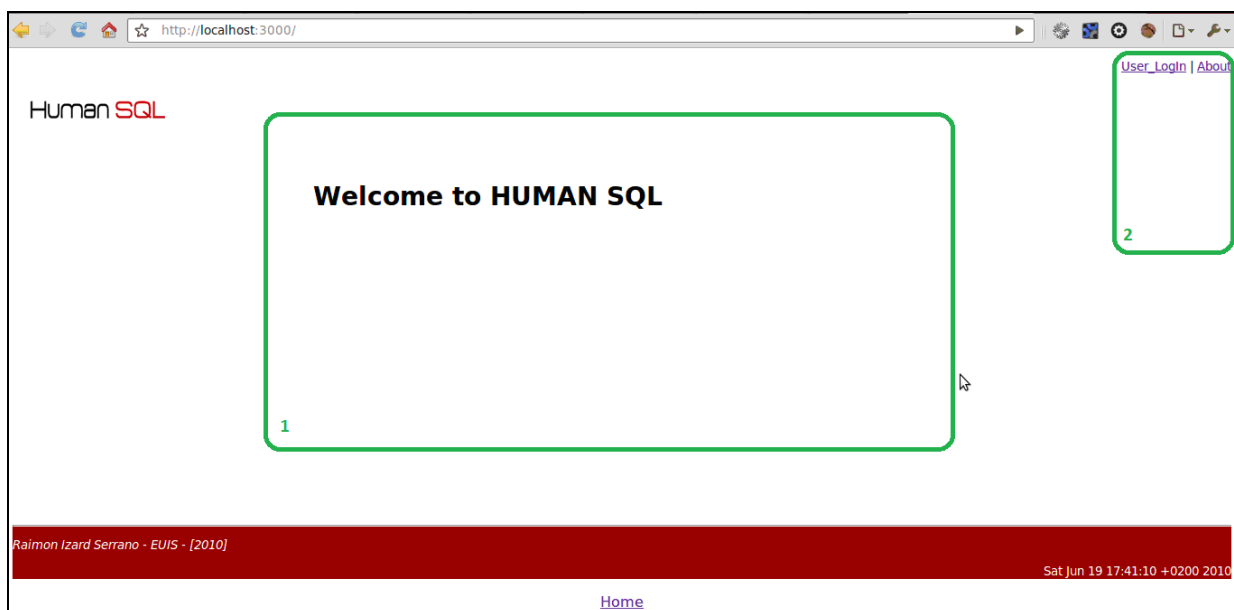


Il·lustració 13: Logotip del producte.

Es fa al·lusió al llenguatge de consultes estàndard *SQL*, i el concepte de llenguatge natural amb la paraula *Human*. Intentant així plasmar com el llenguatge *SQL* és humanitzat en el nostre sistema.

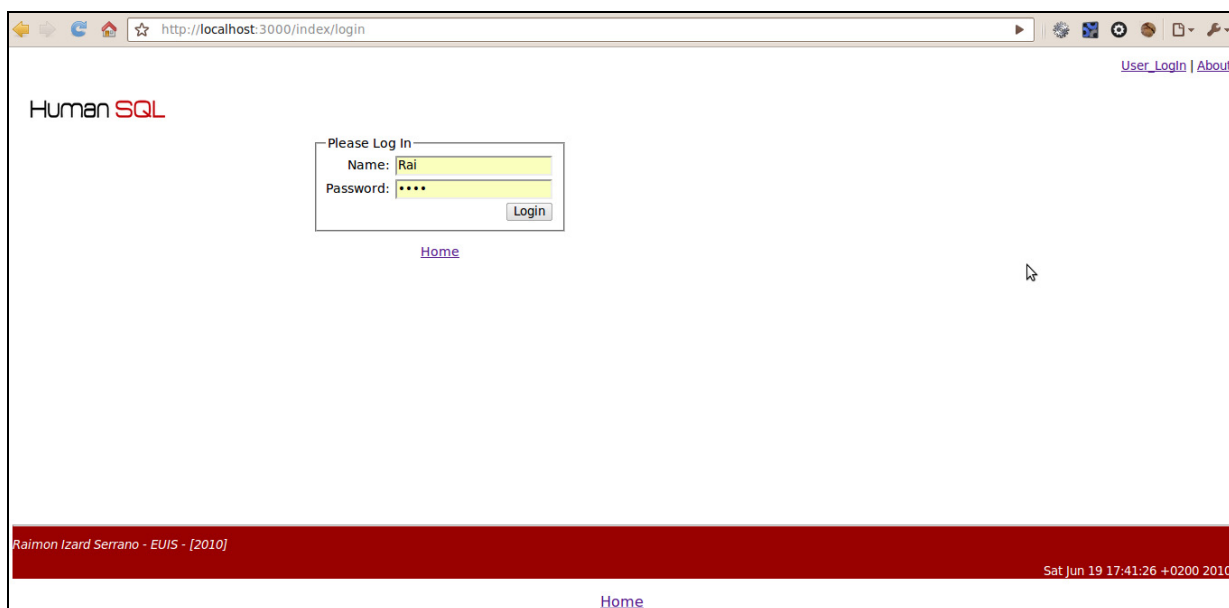
Els objectius de la interfície web són els de crear, editar, eliminar i llistar usuaris i les quatre mateixes accions per a les consultes.

La següent imatge (il·lustració 14) mostra la base de la web on veiem de marcadetes de color verd les àrees que són actives, és a dir on es mostra el contingut del site i la resta es manté estàtica amb l'excepció de l'hora que es va actualitzant. La idea és fer un sistema més ràpid ja que reutilitzem codi i se n'ha de carregar menys a cada transició ja que només es carreguen les àrees marcades.



Il·lustració 14: Imatge de la pantalla de benvinguda de l'aplicació d'un usuari qualsevol amb les parts actives marcades.

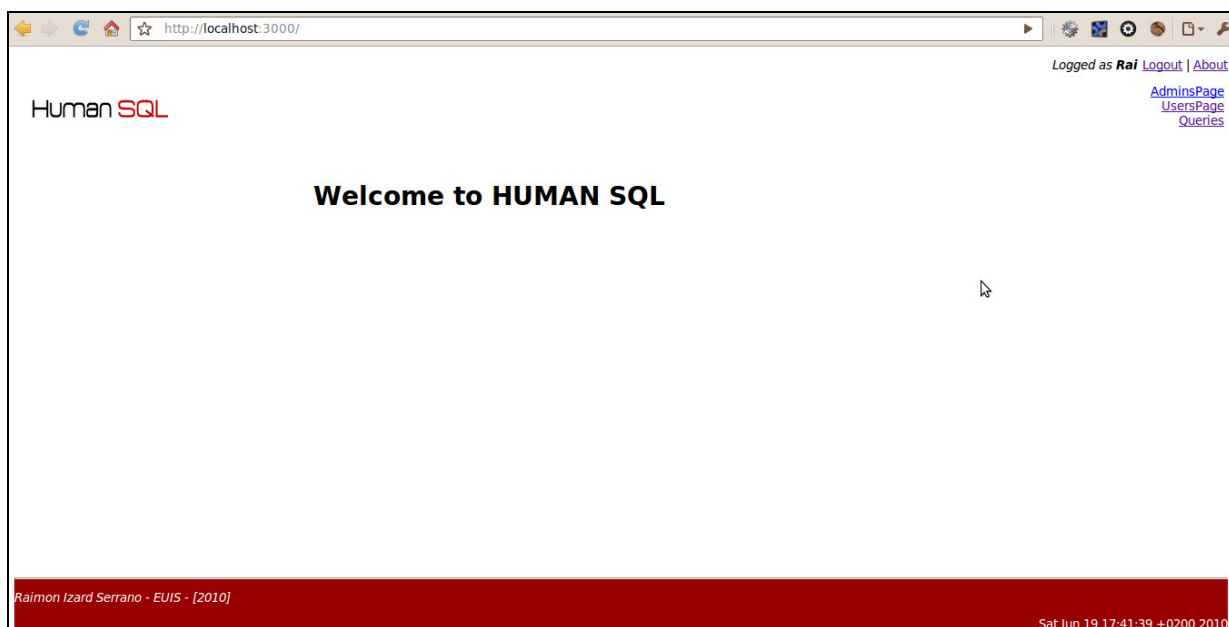
Podem veure a continuació la pantalla principal de log in pels usuaris registrats.



II-lustració 15: Pàgina de log in pels usuaris.

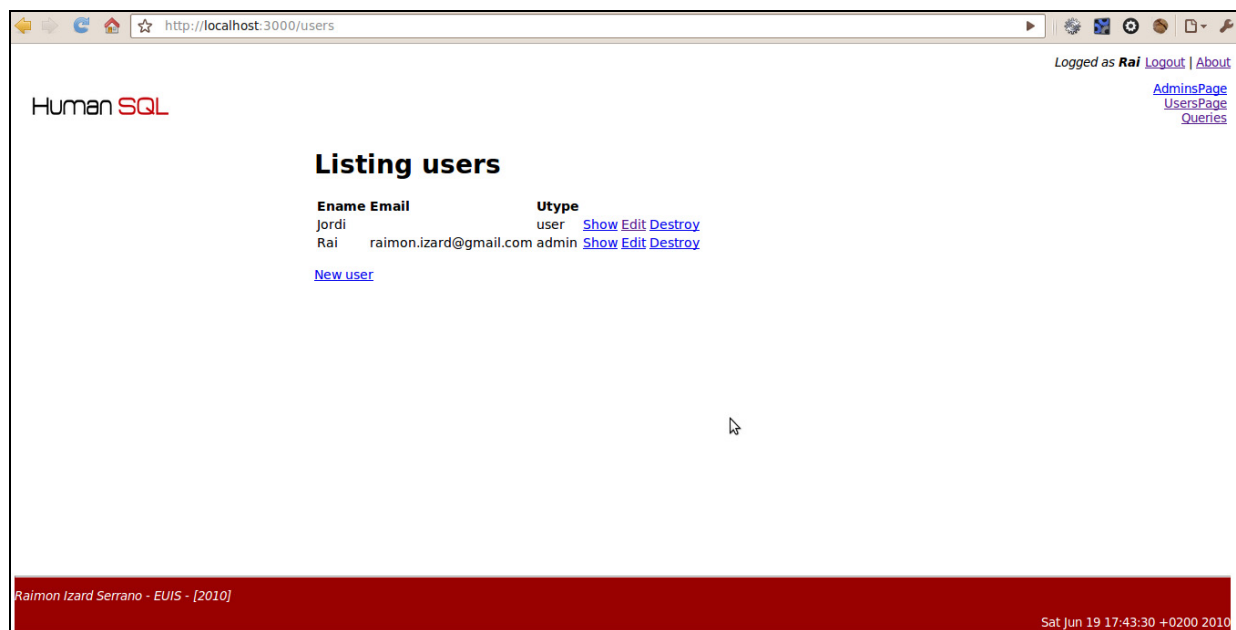
Per crear una compta d'usuari nova primer un administrador s'ha de connectar i crear-lo. Això ens serveix per evitar que qualsevol persona es pugui crear un usuari nou i connectar-se.

A continuació podem veure la mateixa pàgina principal un cop un usuari registrar s'ha connectat al sistema. Podem veure la informació '*Logged as Rai*' que ens indica que estem connectats usant l'usuari *Rai*.



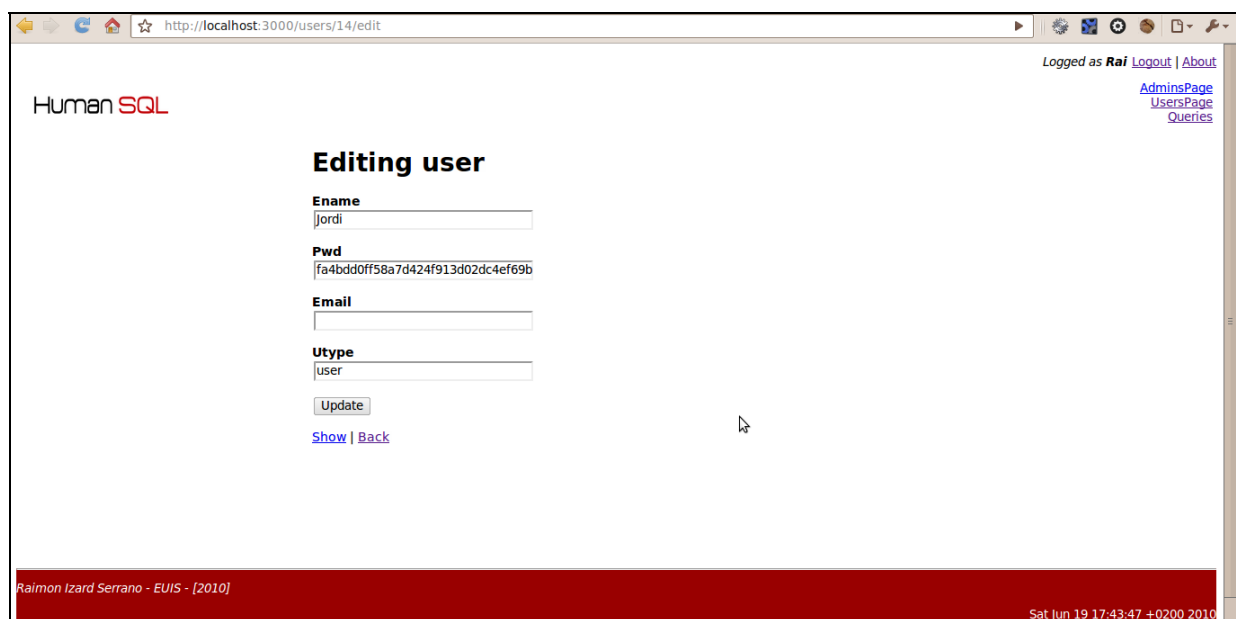
II-lustració 16: Pàgina principal amb usuari connectat.

Vista de la llista d'usuaris del sistema.



Il·lustració 17: Vista de la llista d'usuaris.

Aquí podem veure el formulari d'edició d'un usuari.



Il·lustració 18: Formulari d'edició de les propietats d'usuari.

La següent imatge (Il·lustració 19) ens mostra la vista del llistat de les consultes on podem veure la columna 'Status' que ens indica l'estat de les consulta en cada moment.

User	Ticket	Human	Status	Output	Actions
13	4	tell me admins	Processed	QueryTicket-[4]_at_14-6-2010_104427.csv	Show Edit Destroy
13	5	tell me admins	Processed	QueryTicket-[5]_at_14-6-2010_182559.csv	Show Edit Destroy
13	6	tell me admins	Processed	QueryTicket-[6]_at_14-6-2010_19822.csv	Show Edit Destroy
13	7	How many products are in stock?	Processing	File1	Show Edit Destroy
13	8	How many products are in stock?	Processing	File1	Show Edit Destroy
13	9	How many products are in stock?	Error	File1	Show Edit Destroy

Raimon Izard Serrano - EUIS - [2010] Sat Jun 19 17:44:09 +0200 2010

Il·lustració 19: Vista del llistat de consultes.

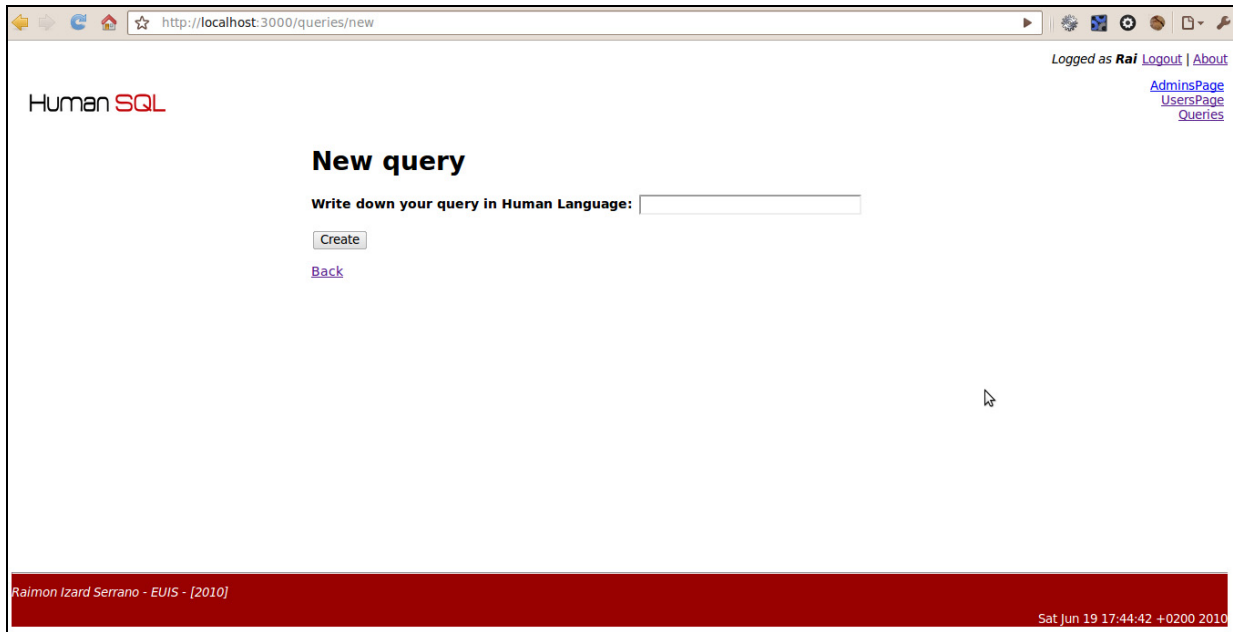
Aquí (Il·lustració 20) podem veure la informació d'una consulta en concret.

Ticket: 4
Human: tell me admins
SQL statment: SELECT * FROM admins
Status: Processed
output: QueryTicket-[4]_at_14-6-2010_104427.csv
[Edit](#) | [Back](#)

Raimon Izard Serrano - EUIS - [2010] Sat Jun 19 17:44:24 +0200 2010

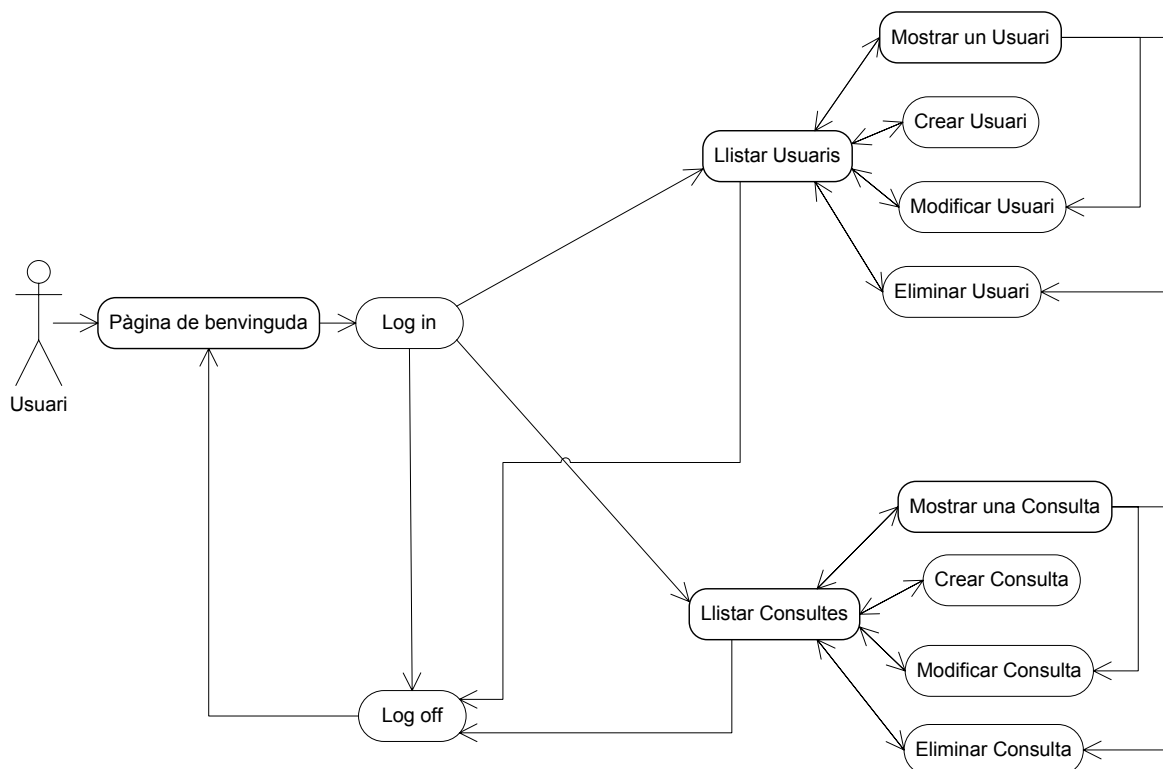
Il·lustració 20: Vista dels detalls d'una consulta.

A continuació (Il·lustració 21) podem veure el quadre de text per inserir una consulta nova en llenguatge natural.



II-lustració 21: Vista de crear una consulta nova.

A continuació podem observar (II-lustració 22) un esquema de tots els camins que es poden explorar entre les diferents vistes o pàgines dins de l'entorn web així com les interaccions entre elles.



II-lustració 22: Arbore de pàgines de l'aplicació web.

3.7. Gestor de consultes

Hem implementat un *script* en Ruby que ens gestiona les consultes usant una solució molt òptima i senzilla que consta de fer ús de la columna '*status*' de la taula de '*queries*' a mode de cua.

Tenim quatre estats definits per a les consultes: *pending* (*pendent*), *processing* (*processant*), *processed* (*processada*) i *error*.

Per controlar quines consultes han de passar per l'analitzador, quines no i en quin ordre, hem implementat una cua on s'executen per ordre d'entrada seguint un algorisme de *FIFO*.

Per a resoldre-ho simplement llancem una consulta a la taula buscant la primera fila començant per dalt de la taula amb la columna *status* = *pending* que correspondrà a la consulta que s'ha introduït fa més temps al sistema i que encara està pendent de ser processada.

Un cop és el torn d'una consulta en concret, és enviada a l'analitzador del llenguatge.

Com podem veure a la (Il·lustració 8) l'analitzador produeix una sortida que el gestor s'encarrega de recollir i la fa servir per actualitzar el camp *sql* de la consulta a la taula *queries*, canvia també el camp *status* a *processing* i s'informa a l'usuari a través de correu electrònic.

Aleshores és quan el gestor també s'ocupa d'enviar la consulta SQL produïda per l'analitzador del llenguatge a la base de dades de lectura del client. Quan aquesta respongui, un cop més el gestor recopilarà les dades i les encapsularà en un fitxer d'extracte en un format adient per a ser posteriorment importat en un full de càlcul si és necessari.

Un cop fet, el camp de *output* s'actualitza amb el nom del fitxer de sortida, també es canvia l'estat de la consulta a *processed* i finalment el gestor envia un correu electrònic a l'usuari propietari de la consulta recentment processada informat del nou estat que serà el de processada.

L'estat serà *Error* només quan la consulta no ha pogut ser validada pel sistema i ha generat una sortida pel bloc de *unknown*. Aleshores, no es llançarà la consulta SQL al sistema, no

s'actualitzarà el camp *sql* a la fila i no es generarà el fitxer de sortida. Però sí que s'enviarà el missatge a l'usuari via correu electrònic.

Totes aquestes accions són fetes pel gestor de consultes, que mou el flux de dades entre els diferents components del sistema i gestiona la cua de peticions pendents. Com es pot observar a la (Il·lustració 7) el gestor interacciona amb tots els mòduls.

Cal comentar que l'*script* del gestor s'executa periòdicament en el servidor on es troba allotjat el sistema usant un *demon* anomenat *crontab*. Es tracta d'un planificador de tasques del sistema operatiu Linux. Cada cop que s'executa el codi es processa una consulta per complet.

La freqüència d'execució es pot modificar canviant la configuració de la tasca planificada. D'aquesta manera es pot ajustar a la potència del servidor per tal de no col·lapsar els recursos físics de la màquina.

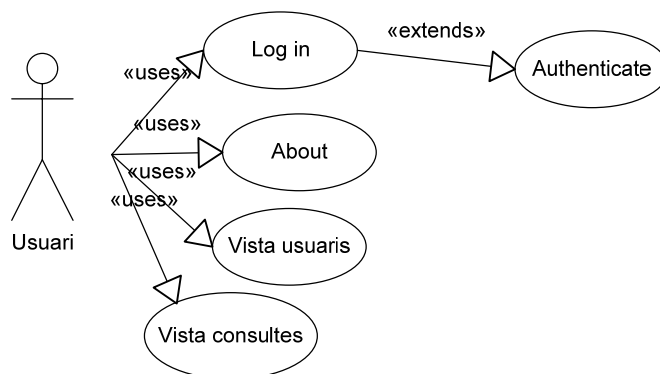
3.8. Diagrames de seqüència, aplicació i casos d'ús

En aquest apartat hem ajuntat tots els diagrames per tal de fer més lleugera la lectura de les seccions anteriors i deixar aquests detalls al final de la part de disseny i implementació per a més informació.

Començarem amb els casos d'ús de les accions més rellevants que poden fer els usuaris (*administrador i usuari senzill*) mitjançant l'aplicació web.

Vista principal:

Mentre els usuaris són a la pàgina principal, només tenen una opció principal a fer que és connectar-se. Un cop fet, poden accedir als links d'usuaris i de consultes. D'altra banda també es pot consultar 'About' per veure els detalls de l'aplicació en qualsevol moment encara que no estiguin connectats.

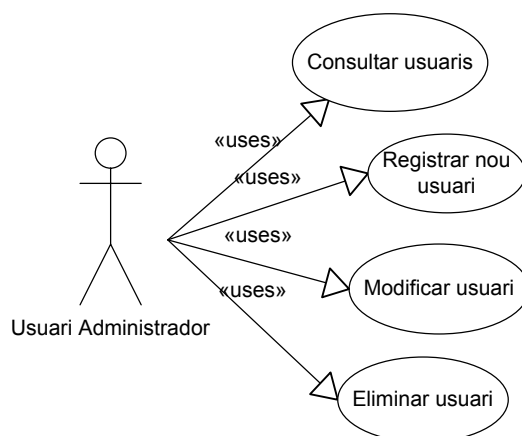


II·lustració 23: Cas d'Ús de tots els usuaris dins de la pàgina principal del site web.

En aquesta interfície no observem diferències entre els usuaris administrador i els usuaris senzills.

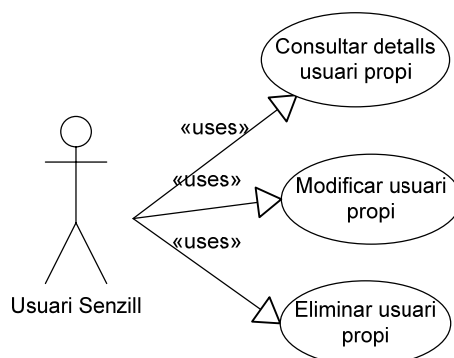
Vista d'usuaris:

Com podem veure, els usuaris amb drets d'administradors, poden consultar, crear, modificar i eliminar tant usuaris senzills com administradors.



Il·lustració 24: CU de l'usuari administrador a la vista usuaris.

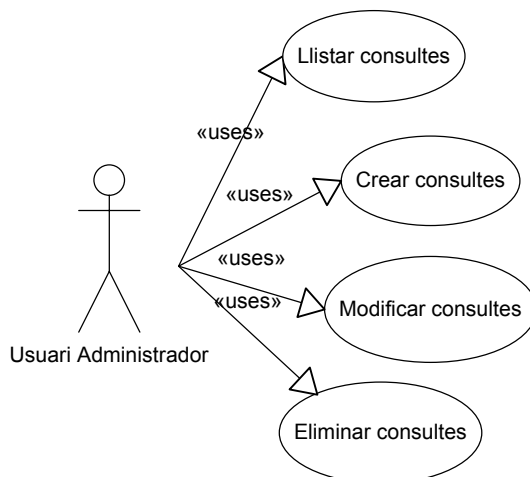
A continuació tenim la mateixa situació plantejada però pels usuaris sense privilegis.



Il·lustració 25: CU d'usuari sense privilegis a la vista d'usuaris.

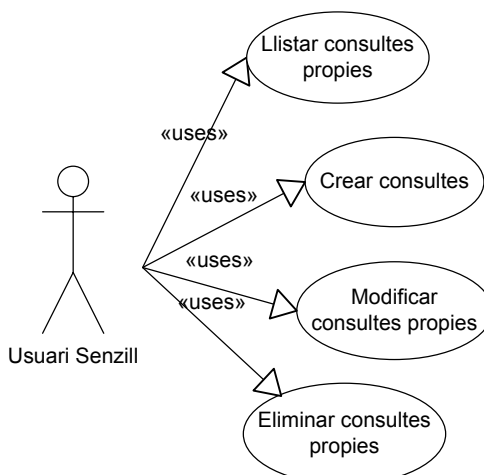
Vista de consultes:

Veiem a continuació les accions disponibles per un usuari administrador dins la vista de consultes.



Il·lustració 26: CU d'administrador a la vista de consultes.

El mateix pot fer l'usuari senzill però només amb les consultes que ell ha creat i de les quals és propietari.



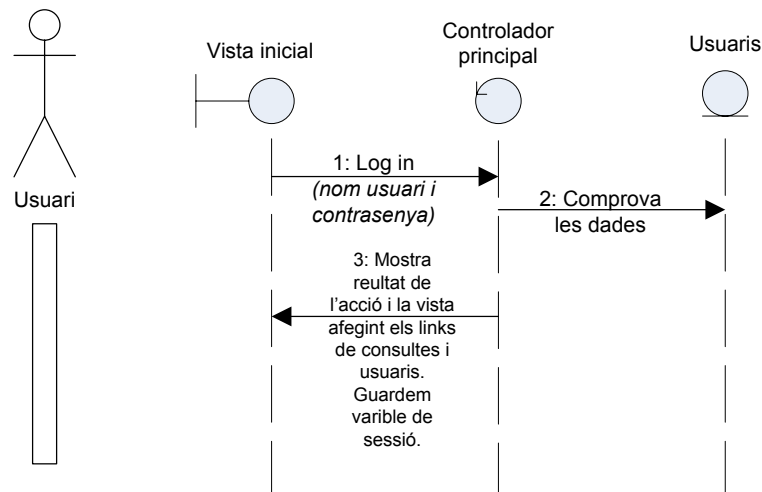
Il·lustració 27: CU d'usuari senzill a la interfície de consultes.

A continuació podem veure els detalls de les funcionalitats principals del sistema amb la relació d'entrades, sortides i accions que es poden dur a terme dins de l'aplicació web així com els diagrames de seqüència per a cada una d'elles.

Accions sobre la vista principal:

Login d'usuaris	
Informació	Els usuaris i administradors presents en el sistema podran connectar-se.
Entrada	L'usuari ha d'introduir el seu nom i contrasenya.
Procés	El controlador principal de l'aplicació comprova si hi ha un nom d'usuari igual que el que s'ha escrit i si el password correspon.
Sortida	Si és satisfactori, l'usuari queda connectat i pot accedir a les vistes d'usuari i de consultes. Així com es guarda una variable de sessió amb la <i>id</i> de l'usuari connectat. A totes les vistes apareix el nom de l'usuari connectat i desapareix l'opció de 'Log in' per deixar lloc a l'opció 'Log out'.

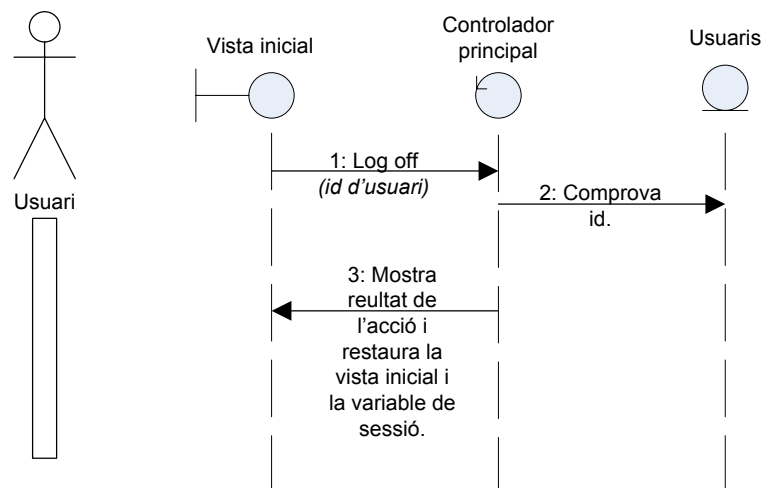
Taula 32: Taula de la funcionalitat de login.



Il·lustració 28: Diagrama de Seqüència de log in d'usuari.

Log out d'usuaris	
Informació	Els usuaris i administradors connectats en el sistema podran desconnectar-se.
Entrada	L'usuari en concret ha de clicar a la icona de log out.
Procés	El controlador principal del sistema comprova la <i>id</i> de la variable de sessió amb l'usuari connectat.
Sortida	Si és satisfactori, l'usuari queda desconnectat i ja no pot accedir a les vistes d'usuari i de consultes. Així com es borra la variable de sessió amb la <i>id</i> de l'usuari connectat. A totes les vistes desapareix el nom de l'usuari connectat i desapareix també l'opció de 'Log out' per deixar lloc a l'opció 'Log in' un altre cop.

Taula 33: Taula resum de la funcionalitat de log off.

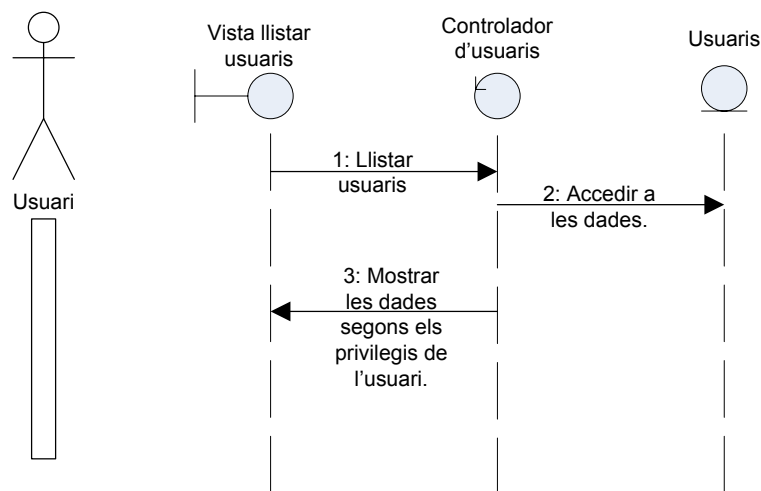


Il·lustració 29: Diagrama de seqüència log off d'usuari.

Accions sobre la vista d'usuaris:

Llistar usuaris	
Informació	Els administradors poden veure tots els detalls de tots els usuaris. Els usuaris senzills només poden consultar els detalls del seu propi usuari.
Entrada	No hi ha entrada de dades ja que es tracta d'una acció de consulta.
Procés	El controlador d'usuaris fa una cerca filtrada segons la situació.
Sortida	Es mostra la informació corresponent de forma ordenada per pantalla.

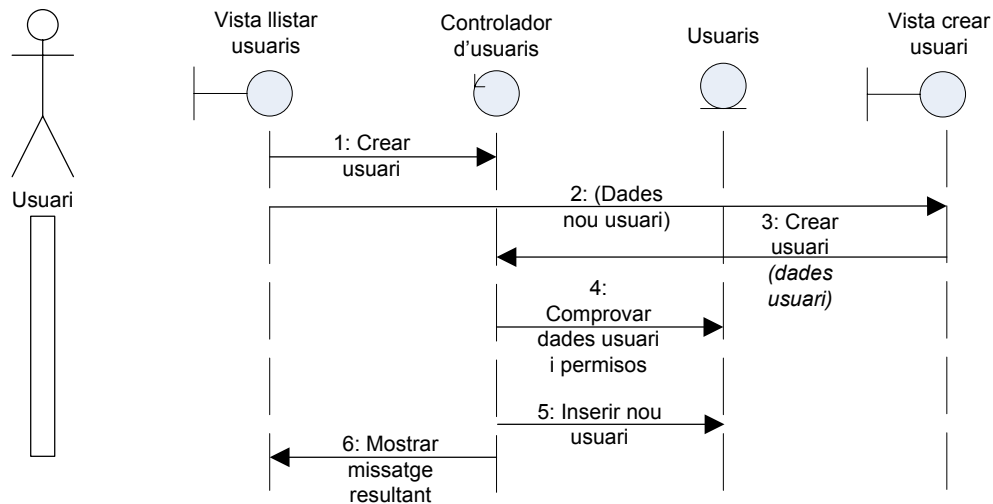
Taula 34: Funció de llistar usuaris.



Il·lustració 30: DS Llistar usuaris.

Crear usuari	
Informació	Només els administradors poden donar d'alta a nous usuaris per tal d'evitar un descontrol d'usuaris al sistema i que persones alienes puguin fer-ne ús sense permís ja que recordem que es tracta d'un sistema via web.
Entrada	L'administrador ha d'omplir els detalls d'usuari tals com nom, password, email, tipus d'usuari. La resta de camps tals com la ip són automàtics.
Procés	El controlador d'usuaris de l'aplicació comprova les dades d'entrada i registra el nou usuari si tot és correcte.
Sortida	Es mostra un missatge per pantalla de que tot ha estat correcte si aquest és el cas, si no es reclamen les dades que falten.

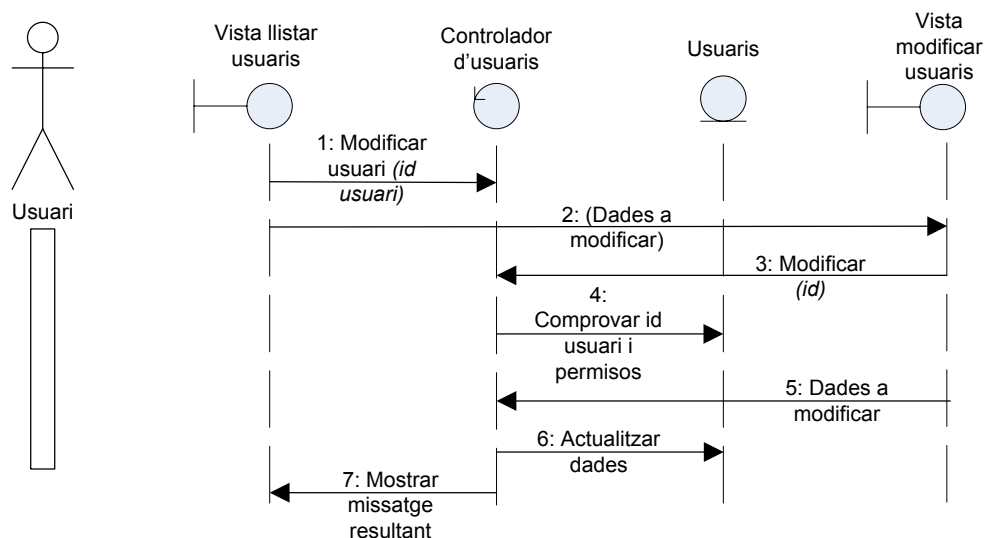
Taula 35: Funció de crear usuari.



II·lustració 31: DS crear usuari.

Modificar usuari	
Informació	Els administradors poden modificar els detalls de tots els usuaris senzills i els propis però no els de la resta d'administradors. La resta d'usuaris només poden editar els seus propis detalls.
Entrada	Es modifiquen els camps desitjats.
Procés	El controlador d'usuaris de l'aplicació comprova les dades d'entrada i efectua els canvis si tot és correcte.
Sortida	Es mostra un missatge per pantalla de que tot ha estat correcte si aquest és el cas, si no es reclamen les dades que falten.

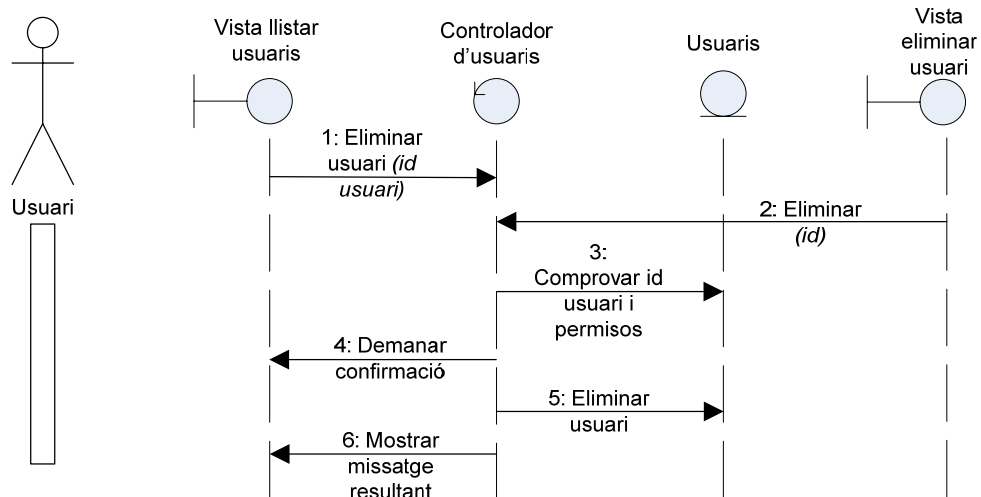
Taula 36: Detalls de l'acció de crear usuari.



II·lustració 32: DS modificar usuari.

Eliminar usuari	
Informació	Els administradors poden eliminar qualsevol usuari exceptuant altres administradors.
Entrada	S'agafa la <i>id</i> d'usuari per tal d'eliminar-lo
Procés	El controlador d'usuaris de l'aplicació llança un missatge d'alerta per tal de confirmar l'acció. S'eliminen les consultes pendents de l'usuari eliminat, però les processades perduren a l'historial. Si es tracta d'un usuari no administrador, evidentment perd la connexió al sistema.
Sortida	S'elimina l'usuari del sistema i es mostra un missatge del resultat de l'acció.

Taula 37: Acció d'eliminar usuari.

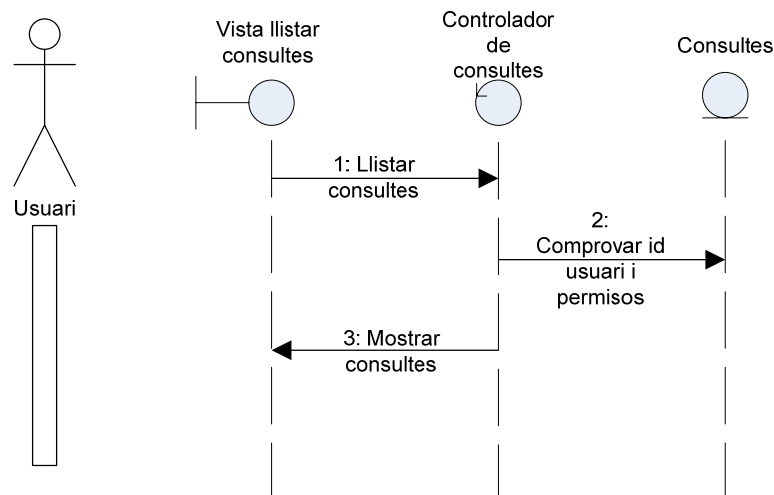


Il·lustració 33: DS Eliminar usuari.

Accions a la vista de consultes:

Llistar consultes	
Informació	Els administradors poden veure totes les consultes de tots els usuaris. Els usuaris senzills només poden consultar els detalls de les seves pròpies consultes.
Entrada	No hi ha entrada de dades ja que es tracta d'una acció de consulta.
Procés	El controlador de consultes fa una cerca filtrada segons la situació.
Sortida	Es mostra la informació corresponent a les columnes del número de tiquet obert per cada consulta, la consulta en llenguatge natural, l'estat de la consulta i el seu fitxer de sortida.

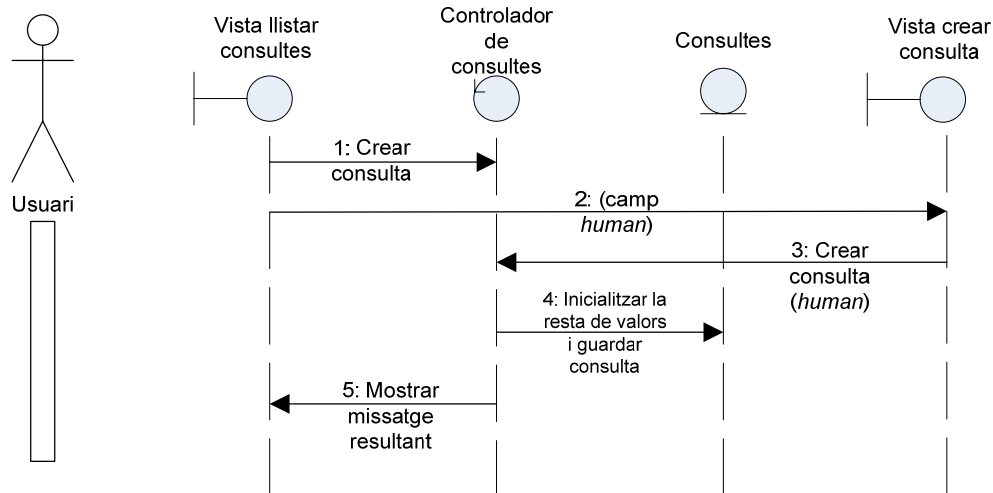
Taula 38: Detalls de l'acció de llistar consultes.



Il·lustració 34: DS Llistar consultes.

Crear consulta	
Informació	Tots els usuaris poden crear consultes noves un cop estan connectats.
Entrada	L'usuari ha d'introduir tan sols la consulta en llenguatge natural en un quadre de text. La resta de camps són només modificables pel sistema.
Procés	El controlador de consultes de l'aplicació comprova la dada d'entrada i registra la nova consulta si tot és correcte a la vegada que es posa a la cua de consultes pendents.
Sortida	Es mostra un missatge per pantalla de que tot ha estat correcte si aquest és el cas, si no es reclamen les dades que falten.

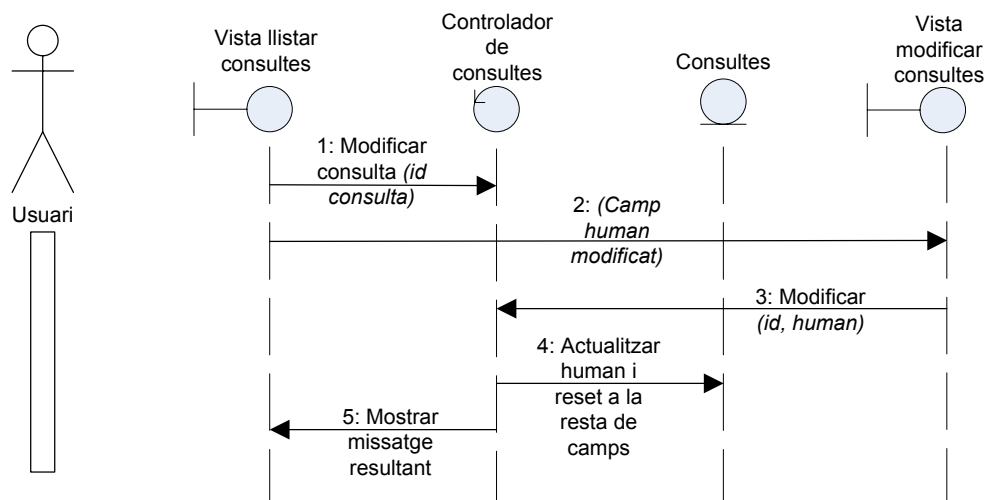
Taula 39: Detalls de crear consulta.



Il·lustració 35: DS Crear consulta.

Modificar consulta	
Informació	Els administradors poden modificar els detalls de totes les consultes. La resta d'usuaris només poden editar les seves pròpies consultes i només el camp de la consulta, no la resta de detalls que són propietat de sistema.
Entrada	Es modifiquen les dades desitjades.
Procés	El controlador de consultes comprova que tot sigui correcte i guarda els canvis a la base de dades.
Sortida	Es mostra un missatge per pantalla de que tot ha estat correcte si aquest és el cas, si no es reclamen les dades que falten.

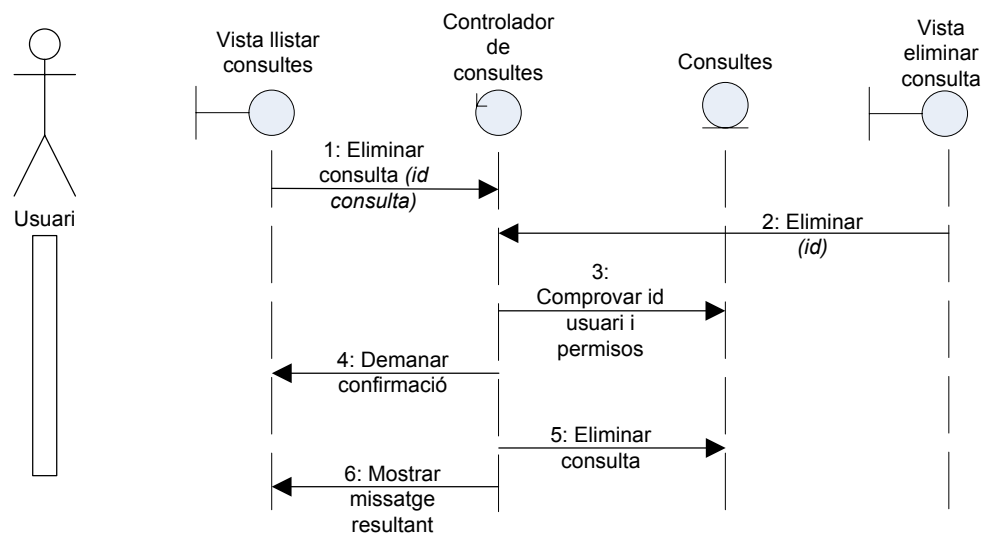
Taula 40: Detalls de l'acció de modificar consulta.



Il·lustració 36: DS Modificar consulta.

Eliminar consulta	
Informació	Els administradors poden eliminar qualsevol consulta i la resta d'usuaris només poden eliminar les pròpies.
Entrada	Es guarda el número de tiquet de la consulta a esborrar que correspon amb el camp <i>id</i> de la taula <i>queries</i> .
Procés	El controlador de consultes de l'aplicació llança un missatge d'alerta per tal de confirmar l'acció.
Sortida	S'elimina la consulta del sistema i es mostra un missatge del resultat de l'acció.

Taula 41: Propietats de l'acció d'eliminar consulta.



Il·lustració 37: DS Eliminar consulta.

4. Fase de proves

Per aquest apartat hem de tenir molt en compte que estem davant d'un projecte on el resultat és un prototip no un producte en versió final.

Aleshores doncs s'han aplicat uns criteris de testos basats estrictament en la funcionalitat del sistema. Per tant, la seguretat del sistema no queda en cap moment garantida.

Es dona per entès que el producte en el seu estat actual no es farà servir sota un entorn de producció si no que només a mode de demostració, situació en la qual garantir la seguretat no és transcendental.

S'han aplicat bàsicament proves d'unitat mòdul a mòdul per comprovar de forma independent que cadascun d'ells funciona correctament.

Paral·lelament, cada cop que un mòdul superava correctament les proves d'unitat per assegurar-ne la funcionalitat en els casos bàsics, s'han anat agrupant en el producte beta final per aplicar proves d'integració pas a pas.

D'altra banda, al no tenir un client que ens hagi encarregat el prototip, les proves d'acceptació han estat validades per l'equip de desenvolupadors.

4.1. Proves d'unitat de l'analitzador del llenguatge i la gramàtica

A la següent taula tenim les proves principals a les quals hem sotmès l'analitzador del llenguatge natural.

#	Requeriment	Resultat esperat	Resultat real	Tester
1	Integrar la llibreria <i>treetop</i> dins del sistema.	Instal·lar correctament el <i>plugin</i> i usar les seves utilitats per consola.	Com l'esperat.	Raimon
2	Validar sentències aïllades.	Dissenyar una gramàtica capaç de validar les frases d'entrada.	Com l'esperat.	Raimon
3	Producció de sortida per qualsevol entrada. Incloent símbols no alfabètics.	Que el nostre sistema sigui determinista i que consumeixi qualsevol entrada encara que no pugui ser validada.	Com l'esperat.	Raimon

Taula 42: Taula de proves sobre l'analitzador de llenguatge.

4.2. Proves d'unitat de la BD

A continuació podem veure una taula dels testos principals als quals hem sotmès la Base de Dades:

#	Requeriment	Resultat esperat	Resultat real	Tester
1	Accés complet a les taules del sistema a través de phpMyAdmin.	Poder connectar a la BD a través del navegador Google Chrome a la direcció <i>localhost/phpmyadmin</i> a dins de Ubuntu 9.10.	Com l'esperat.	Raimon
2	Poder executar sentències SQL des de l'interfície de phpMyAdmin.	Executar select, insert, update i delete.	Com l'esperat.	Raimon
3	Diferents usuaris amb diferents privilegis a la BD.	Crear un usuari de lectura per Jordi Duran.	Correcte.	Raimon
4	Canviar l'estructura de la BD a través de Ruby on Rails.	Executar migracions de codi RoR sense problemes i crear i modificar taules de la BD.	Correcte.	Raimon
5	Llançar sentències SQL des de RoR.	Executar scripts Ruby amb consultes a la BD.	Correcte.	Raimon.
6	Llançar sentències SQL des de la consola de Ubuntu 9.10	Assignar les dades resultants a variables Ruby i poder-hi accedir de forma estructurada.	Correcte.	Raimon.
7	Exportar tot el contingut de la BD per a còpies de seguretat.	Obtenir un fitxer de sortida amb tota l'estructura de la BD per poder importar-la en cas de desastre.	Extracció correcte. Importació també testejada i correcte.	Raimon
8	Llegir l'estructura completa de la BD usant scripts Ruby.	Obtenir un fitxer XML amb l'estructura de la BD usant Ruby.	Correcte.	Raimon
9	Extreure dades de la BD usant scripts Ruby.	Exportar resultats de consultes de la BD a fitxers .csv usant Ruby.	Correcte.	Raimon.

Taula 43: Relació de les proves aplicades sobre la BD.

4.3. Proves d'unitat del l'aplicació web

L'objectiu principal de l'entorn web és que fos un sistema funcional, senzill i ràpid. La resta d'aspectes no menys importants com la seguretat i el disseny no jugaven un paper important en el nostre desenvolupament.

#	Requeriment	Resultat esperat	Resultat real	Tester
1	Executar el servidor del site web en local.	Poder accedir a la web usant un navegador a la direcció <i>localhost:3000</i> .	Correcte	Raimon
2	Executar el servidor web dirigint el site a una ip concreta per ser accedida per tercers.	Accedir al site usant un altre equip des d'un país estranger.	Correcte.	Raimon i col·laboradors externs.
3	Site web compatible amb diferents SO.	Obtenir una imatge de disseny gràfic de la web inamovible i independent del SO.	No superat. El site web ha estat desenvolupat i testejat sota Ubuntu 9.10. i funciona correctament sota aquest entorn. Però té un resultat acceptable en Windows 7, té més problemes en Windows XP i no ha estat provat en sistemes MACOS. Cal comentar que els problemes venen donats per l'ús de navegadors que no siguin Chrome.	Raimon i Jordi Duran.
4	Accedir al site des de qualsevol navegador.	Obtenir la mateixa interfície gràfica independentment del navegador usat.	No superat. Funciona perfectament en la majoria de SO sempre que s'usi Google Chrome. Resultats molt	Raimon, Jordi Duran.

			negatiu en la majoria de versions de IE. Resultats acceptables en FireFox.	
5	Proporcionar un sistema de log in i log out funcional	Establir diferents permisos i nivells d'usuari així com una vista inicial de benvinguda que obligui a connectar-se per seguir.	Correcte.	Raimon.
6	Prohibir l'accés a zones restringides del sistema sense estar connectat.	No mostrar links de les vistes, bloquejar les url directes, control de permisos i de variables de sessió.	Correcte.	Raimon.
7	Seguretat d'entrada de dades per formularis.	Impunitat contra tot tipus d'atacs que puguin perjudicar la integritat de les dades de la BD.	No superat. Hem aplicat alguns filtres en la majoria de formularis, però no assegurem que el site estigui blindat, ja que en cap moment ho hem garantit ni ha estat un objectiu de projecte ja que es tracta d'un prototip.	Raimon.
8	Proveir accés per a dispositius mòbils.	Que es pugui navegar per la web usant un mòbil.	Accés possible, però la pàgina web no està preparada per a pantalles petites. Es pot usar mitjançant el mòbil però la satisfacció és baixa ja que es desenquadra tota la web.	Raimon.

Taula 44: Detalls de les proves aplicades sobre el site web.

4.4. Proves d'unitat del gestor de consultes

Resum de les proves aplicades sobre el gestor:

#	Requeriment	Resultat esperat	Resultat real	Tester
1	Implementar correctament la cua seguint un ordre FIFO.	Que sempre agafi la consulta pendent adient.	Com l'esperat.	Raimon
2	Comunicació correcte amb l'analitzador del llenguatge.	Que enviï la consulta i en reculli la sentència SQL produïda.	Com l'esperat.	Raimon
3	Connexió de lectura amb la BD del client.	Llançar les consultes SQL produïdes i recollir-ne les dades resultants.	Com l'esperat.	Raimon
4	Encapsular les dades en un fitxer d'extracte.	Crear un fitxer de sortida amb les dades en un format adient separant els camps entre el caràcter ';'.	Com l'esperat.	Raimon
5	Enviar correus electrònics als usuaris.	Informar als usuaris dels canvis d'estat de les peticions.	Com l'esperat.	Raimon
6	Actualitzar les dades de les peticions dins de la BD del sistema.	Canviar l'estat de les consultes i omplir els camps de <i>output</i> i <i>sql</i> .	Com l'esperat.	Raimon

Taula 45: Proves d'unitat sobre el gestor de consultes.

5. Conclusions

En aquest capítol finalitzarem amb l'avaluació dels objectius un cop finalitzat el projecte així com amb un llistat de futures línies de treball per al prototip, repassarem les desviacions de la planificació i exposarem una valoració personal de l'experiència.

5.1. Valoració d'objectius assolits

El nostre objectiu principal era el d'oferir la possibilitat d'accedir a la informació emmagatzemada dins d'una base de dades a usuaris no experts.

Un cop fet el prototip que ofereix la funcionalitat esperada i aconsegueix els objectius marcats podem afirmar que és viable desenvolupar aquesta eina.

Si anem al detall i repassem els objectius marcats dins de l'estudi de viabilitat (*apartat 2.4.1*), veiem que consta de 10 objectius:

1. Consultar i obtenir informació, de manera accessible per a l'usuari no expert, d'una base de dades usant un subconjunt del llenguatge natural de la Llengua Anglesa. (**Complert**)
2. Gestor web de tiquets per a la monitorització automàtica de l'estat de les consultes. (**Complert**)
3. Sistema configurable i amb capacitat d'actualització per tal d'interpretar un subconjunt més ampli de la llengua en el futur. (**Complert**)
4. Gestió de la informació de l'estat de les consultes i de l'entorn via correu electrònic. (**Complert**)
5. Compte d'administrador pel manteniment de la BD i el sistema. (**Complert**)
6. Encapsulament dels resultats en fitxers en format .csv (**Complert**)
7. Subministrar serveis web per a poder reutilitzar les funcions del nostre sistema en programari de tercers. (**No complert**)

8. Addicionalment dissenyar una aplicació portable executable en local usant les crides a web services. (**No complert**)
9. Eina d'autocompletat de camps i/o noms de taules dins de la interfície d'usuari. (**No complert**)
10. Manuals d'usuari i administrador. (**Complert**)

Tal com podem observar, els objectius 7, 8 i 9 no els hem portat a terme pel fet de sobrepassar la dedicació esperada per al projecte. Com es pot observar a (*Taula 1: Taula de catalogació d'objectius.*), aquests tres objectius van ser catalogats com a secundaris per tant el greuge de no haver-los desenvolupat no és transcendental ja que no recau en la funcionalitat que es cercava en el projecte.

El producte comercial que se'n pot derivar és factible perquè el nostre prototip així ho justifica.

5.2. Futures línies de treball

Al tractar-se d'un sistema molt modular i d'estar desenvolupat en Ruby on Rails, ofereix grans possibilitats en el desenvolupament web.

Treballar durant aquest temps en Ruby ens ha fet adonar de que pràcticament tot és possible usant aquest llenguatge de programació. Ens ha proporcionat l'avantatge de poder resoldre funcionalitats complexes de manera clara i senzilla. El codi font està pensat per ser codi lliure, i per tant, ofereix la possibilitat de que hi hagi una comunitat de desenvolupadors que hi puguin treballar per anar millorant les funcionalitats ja que tot els mòduls poden ser modificats sense greuges importants i les proves d'integració no es veurien afectades en gran part.

A continuació podem veure un llistat de les funcionalitats i extres que s'han fixat com a futures línies de treball. Estan numerades alfabèticament perquè no s'ha establert un ordre de prioritats de desenvolupament perquè estaria en funció d'un possible client. En aquests moments no està planificat portar-les a terme ja que se n'hauria de fer un nou pressupost i estudi de viabilitat adient.

- a. Afegir les funcionalitats de Web Services que ens ha quedat pendent en el desenvolupament. Subjecte a que es desenvolupi un programa client que ho requereixi.
- b. Desenvolupar una versió per execució en local.
- c. Sistema d'edició més complex que permeti l'autocompletat de consultes que ofereixi alternatives quan una petició no pot ser processada i validacions parcials.
- d. Ampliar la gramàtica per tal de que sigui capaç de processar consultes en llenguatge natural més variades.
 - d.1. Afegir sinonímia al lèxic.
 - d.2. Flexibilitat per seqüències de text que no coincideixin.
 - d.3. Accions no només de consulta si no també d'edició de les dades.
- e. Carregar consultes en llenguatge natural de forma massiva des de fitxers.
- f. Adaptar el fitxer de sortida a les necessitats del client.
- g. Enviar el fitxer de l'extracte de dades adjunt en el correu electrònic de notificació sempre i quan la mida del fitxer així ho permeti.

- h. Usar la informació del temps d'execució de les consultes per tal d'informar a l'usuari quan temps ha tardat la seva consulta si ja ha estat processada o per contra quan de temps li falta.
- i. Assegurar la seguretat del site web.
- j. Generar logs i reports del sistema sobre l'estat de la base de dades, les consultes en execució, el nombre de consultes pendents, els usuaris connectats, la ratio d'encert en validació de les consultes...
- k. Encapsular el prototip en un producte final.

5.3. Seguiment de la planificació

Les planificacions de projectes complexos sempre tendeixen a ser optimistes i no s'acaba complint amb plans establerts al inici. En concret, la planificació d'aquest projecte s'ha vist modificada respecte la que es va projectar als voltants de Novembre de l'any anterior.

Hi ha molts aspectes que han intervingut en el desfasament de varies etapes, però cal dir que des de l'inici s'havia assumit que la planificació presentada dins de l'estudi de viabilitat estava condicionada a complir amb les hores de dedicació esperades pel projecte de final de carrera i d'aquesta manera va resultar massa optimista.

Cal comentar que a finals de Gener ens va aparèixer un dels riscos més desagradables que és el d'accident en l'eina de treball tal i com està contemplat a (*Taula 22: Relació de riscos.*) al número 5.

D'altra banda, el projecte va sorgir d'una idea, per resoldre un problema experimentat dins del món laboral. Aleshores amb el pensament d'elaborar una proposta d'innovació per un possible producte nou, es va fer expressament de no posar limitacions al llistat de requeriments. Hem estat conscients de que el nombre d'hores destinades al PFC no permetria acomplir-les totes. Però hem volgut posar un escenari real en què un enginyer es pot trobar.

A continuació podem veure una relació de problemes de desenvolupament que ens hem anat trobant al llarg del projecte on hem hagut de fer un esforç extra per resoldre'ls.

Problema	Solució aplicada
Problemes amb el llenguatge Ruby i el seu framework Rails deguts a la desconexió del llenguatge de programació.	Constant aprenentatge i recerca d'informació al web, tutorials, llibres, etc.
Executar el servidor de la web a través de la xarxa distribuïda d'Internet i no només dins de la xarxa local.	Aquest va ser un gran problema que ens va portar uns quants dies fins que provant configuracions del sistema operatiu i de Rails vam aconseguir solucionar-lo.
Problema físic al disc dur del portàtil principal del desenvolupador.	Ens va obligar a migrar tot el projecte en un altre portàtil i a usar la màquina virtual freeware VMWare.
Es volia un sistema adaptable a qualsevol entorn. Per tant, ens havia de permetre llegir l'estructura de la base de dades de lectura del client en cada cas per tal de el sistema conegués quines taules i columnes contenia per tal de validar-ne els noms a les consultes en llenguatge natural.	Es va haver de dissenyar un script que llegís la informació de la base de dades i generés una gramàtica a mida en cada cas amb la informació corresponent.
Problema greu en el desenvolupament de la gramàtica que no ens permetia produir les sentències SQL equivalents al mateix temps de validar les consultes en llenguatge natural.	Partíem de desconexió total de la llibreria <i>treeTop</i> de Ruby. Ens vam veure obligats a començar de zero i aprendre a fer-la funcionar degudament tot i que no hi ha gaire documentació al tractar-se d'una llibreria innovadora.
Extreure les dades en un format correcte. Sabíem com llançar les consultes SQL generades però vam tenir problemes per extreure la informació i encapsular-la en fitxers en el format correcte.	Vam aconseguir processar la variable on es retornava tot el bloc d'informació de la base de dades i desgloçar-la en diferents vectors per tal de poder imprimir el fitxer correctament separant els camps pel caràcter ';'.
Problema al enviar correus electrònics, ja que cada cop que la connexió a Internet canviava, havíem de reconfigurar el mòdul	Ho vam resoldre instal·lant <i>postfix</i> . Una funcionalitat del sistema operatiu Linux que permet configurar el teu propi servidor de

d'enviament de correus amb el nou nom de servidor de correu SMTP.	correu en local. Aleshores el sistema és capaç d'enviar correus independentment de la seva connexió a Internet.
Executar el flux de processament de consultes periòdicament.	Es va resoldre usant el demon anomenat <i>crontab</i> . Es tracta d'un programa de Linux que serveix per planificar tasques.

Taula 46: Relació de problemes i solucions aplicades.

Entrant en els detalls d'anàlisi del seguiment de la planificació, a la següent taula (Taula 47), trobem una relació de temps i costos addicionals deguts a demora en el desenvolupament del projecte en comparació amb la planificació inicial feta al Novembre de 2009 que hem pogut veure a l'apartat 2.24.3.

WBS	Nom	Inici	Final	Duració*	Duració extra*	Complet	Cost (€)	Cost extra (€)
1	Fixar els objectius del projecte.	Sep 21	Nov 28	61d 1h	20d	100%	3985.22	+918
1.1	Analitzar la necessitat del mercat actual i avançar-nos al client.	Sep 21	Sep 23	3d	-	100%	300	-
1.2	Estudiar el sector afectat per la necessitat.	Sep 21	Oct 18	6d	-	100%	240	-
1.3	Buscar solucions dins del mercat actual.	Sep 21	Sep 23	3d	-6d	100%	220	-460
1.4	Pluja d'idees.	Sep 23	Oct 17	1d	-	100%	120	-
1.5	Plantejar una proposta de producte per solucionar la manca.	Oct 18	Oct 24	6d	-3d	100%	583.56	-296.11
1.6	Fixar objectius, extrems i límits.	Oct 18	Oct 20	1d	-	100%	83.78	-
1.7	Estudiar la possible acceptació del producte per part del client.	Oct 19	Oct 20	1h	-	100%	56	-
1.8	Estudi de viabilitat.	Oct 19	Nov 28	41d	+29d	100%	2381.89	+1674.12
2	Fixar la base de recursos del projecte.	Nov 28	Dec 4	10d 1h	-6d 22h	100%	476.67	-303.33
2.1	Marcar els suports que necessitem.	Nov 28	Nov 28	1d	-2h	100%	96.67	-23.33
2.2	Anàlisi de sistema Operatiu.	Nov 28	Nov 29	3h	+2h	100%	20	-
2.3	Programari requerit, el seu cost i adquisició.	Nov 28	Nov 29	2d	-4d	100%	80	-160
2.4	Escollir i configurar servidor web.	Nov 28	Dec 4	7d	-3d	100%	280	-120
3	Configurar el sistema de Base de Dades.	Dec 4	Dec 17	17d 7h	+6h	100%	789.78	-
3.1	Dissenyar el model relacional, diagrama entitat-relació.	Dec 22	Gen 9	3d	-	100%	120	-

3.2	Analitzar els diferents motors de BD disponibles i compatibilitats.	Dec 22	Dec 29	1d	-	100%	40	-
3.3	Fer realitat el disseny a través de les eines de Ruby dins del motor de BD.	Gen 9	Gen 14	6d	-	100%	240	-
3.4	Accés i manteniment de a la BD a través d'algun gestor de BD.	Dec 22	Gen 9	3h	+2h	100%	20	-
3.5	Graella de tests de la BD així com estudis d'escalabilitat i multi-usuari.	Gen 14	Gen 19	7d 4h	+4h	100%	369,78	-
4	Desenvolupar Site Web.	Nov 30	Mai 01	95d	+41d	100%	3782.61	+1565.29
4.1	Estudiar Framework Rails.	Nov 30	Des 8	10d	-	100%	400	-
4.2	Configurar el sistema, dissenyar models de dades.	Des 12	Des 12	1d	-	100%	40	-
4.3	Gestionar la BD a través de Rails.	Des 12	Des 14	3d	-	100%	120	-
4.4	Generar vistes per a cada model.	Des12	Gen 12	20d	+17d	100%	800	+680
4.5	Enllaçar vistes.	Gen 12	Gen 24	5d	+4d	100%	200.08	+160.01
4.6	Crear menú principal.	Gen 12	Gen 16	3d	+2d	100%	120.42	+80.28
4.7	Aplicar estil.	Gen 16	Gen 18	4d	+1d	100%	160	+40
4.8	Rols d'usuari.	Des 14	Des 17	3d	-	100%	120	-
4.9	Implementar seguretat.	Des 17	Des 21	5d	-	100%	200	-
4.10	Aplicar funcionalitat per afegir i gestionar consultes en PLN.	Dec	Feb 14	15d	+12d	100%	600	+480
4.11	Gestionar fluxos de dades, extracció de dades de la BD i sortides a fitxer.	Mar 08	Mar 10	3d	-	100%	120	-
4.12	Afegir autocompletat de paraules camp i noms de taula.	Mar 10	Mar 11	1d 1h	-	0%	60	-
4.13	Implementar gestor de tiquets de consultes.	Mar 11	Mar 20	12d	+5d	100%	480	+200
4.14	Configurar enviament automàtic de correus electrònics a l'usuari.	Apr 26	Mai-01	5d	+3d	100%	200	+120
4.15	Múltiples peticions d'usuaris.	Dec 17	Dec 19	1d	-	100%	40.06	-
4.16	Graella de testos.	Mar 08	Mar 10	2d	-3d	100%	122.06	-195
5	Motor de PLN.	Mai 02	Jun 6	30d	-2d		2673.64	-40.23
5.1	Dissenyar a nivell teòric la gramàtica i el seu autòmat.	Mai 2	Mai 14	5d	+1d	100%	1364	+79.67
5.2	Implementar la gramàtica en Ruby de forma modular i ampliable.	Mai 14	May 20	9d	+2d	100%	360	+80
5.3	Definir un estàndard de dades a tractar (XML, JSON...)	Mai 25	Mai 28	1d	-	100%	40	-
5.4	Tractament dels fitxers amb l'estàndard escollit en solitari.	Mai 29	Mai 30	2d	-5d	100%	80.11	-199.89
5.5	Inserir el motor dins del sistema complex de la	Mai 29	Jun 02	7d	-	100%	425	-

	BD, SiteWeb.							
5.6	Graella de Testos.	Jun 02	Jun06	4d	-	100%	198	-
5.7	Fer accessible el motor a peticions externes a través de WebServices.	Jun 02	Jun 04	2d	-	0%	205.53	-
6	Desenvolupament d'aplicació executable en local.	Jun 10	Jul 15	13d	-	0%	520	-
6.1	Crear un executable que es proveeixi de les llibreries subministrades pel WebService del sistema.	Jun 10	Jun 22	7d	-	0%	280	-
6.2	Graella de testos.	Jun 22	Jul 03	3d	-	0%	120	-
6.3	Manual d'usuari.	Jul 03	Jul 15	3d	-	0%	120	-
7	Documentació de disseny, annexos, manuals i altres.	Sep 21	Abr 17	50d	+35d	100%	2000	+1400
TOTAL:					+90d 12h		14226.56	+3539.73

Taula 47: Relació de temps i costos addicionals debuts a demora en el desenvolupament del projecte.

**Cal tenir en compte que la durada de les etapes en dies i hores es basen en dies laborables marcats pels calendaris de cada actor que ha intervingut en la elaboració de les tasques, no per dies naturals.*

La desviació és moderada ja que sobre els 10687€ pressupostats a l'inici en concepte del cost total de les activitats, s'ha augmentat un 33,12% que és el que representen els 3539.73 euros de més sobre el cost total de les activitats pressupostat al Novembre.

Aleshores el cost final del projecte un cop portat a terme suma un total de:

- Desenvolupament de les activitats del projecte: 14226.56 €
- Costos d'amortització de material: 585.63 €
- Altres costos: 220 €

Total: 15032.19 €.

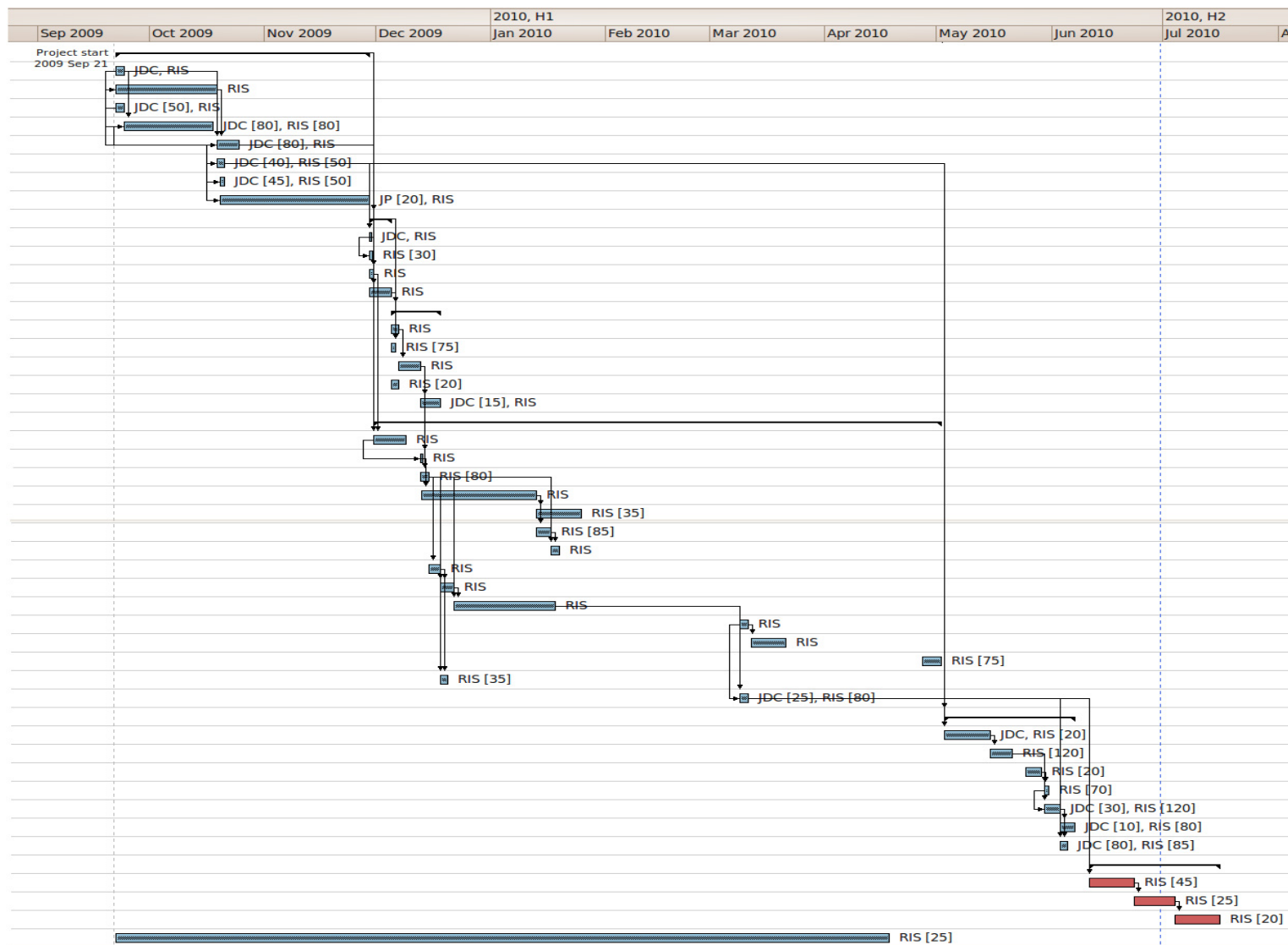
El preu final és elevat i es desvia força del pressupostat a l'inici (11492,46€), però tenint en compte les característiques del projecte i que hem desenvolupat un prototip innovador que requeria d'investigació és encara un preu correcte. Aleshores doncs, clarament vem realitzar una planificació optimista que ha estat modificada pels riscos que hem anat trobant al llarg del desenvolupament. També cal tenir en compte que, com ja hem comentat, la planificació feta al Novembre es va veure condicionada pel fet de no superar les 200 hores de dedicació esperades pel projecte.

En termes temporals, havíem planificat un final de projecte pel 23 de Gener del present 2010 i s'ha demorat fins a tal punt que hem superat la data límit del 29 de Juny de 2010 fet que ens ha obligat a prescindir de portar a terme d'algunes etapes tals com l'activitat 4.12 que pretenia que el sistema proposés paraules per autocompletar les consultes en llenguatge natural dels usuaris amb els noms de taules i/o columnes de la base de dades, l'activitat 5.7 dedicada a subministrar webservices i el bloc d'activitats 6 per complet que pretenia oferir una demo instal·lable en local tal com estava establert per l'objectiu de projecte número 8.

Excloent aquestes activitats, la data final del projecte queda fixada pel 29 de Juny de 2010, dia d'entrega. Per tant doncs, em comès una demora de pràcticament 5 mesos a la qual se li poden descomptar els mesos de Febrer i Abril ja que per incompatibilitat horària i motius pràctics no va ser possible dedicar-hi les hores de treball establertes.

Cal comentar que els 90 dies de més que mostra la taula, són sense tenir en compte les tasques que es podien desenvolupar en paral·lel, per tant, no són 90 dies naturals sobre la data final planificada al Novembre de 2009.

Podem veure a la següent pàgina, (Il·lustració 38), el diagrama de Gantt elaborat un cop realitzat el projecte. Per tant doncs, aquest diagrama s'ajusta al desenvolupament real que hem seguit des del Setembre del 2009. Podem observar en comparació amb el que es va planificar al Novembre de 2009, (Il·lustració 4), com aquest segon diagrama de Gantt és més realista i viable. Aleshores doncs, ens reafirmem en que el primer que es va fer no s'ajustava correctament i la seva planificació associada era massa optimista.



Il·lustració 38: Diagrama de Gantt del projecte un cop desenvolupat.

5.4. Valoració personal

A nivell personal cal remarcar que des de l'inici vaig plantejar aquest projecte no com a exercici acadèmic de final de carrera si no com a una prova definitiva de les meves capacitats i habilitats sobre un escenari el més pròxim possible al món laboral.

La idea del tema de projecte va sorgir durant la meva estada a la consultora *Accenture SL*.

A partir d'aquesta idea, juntament amb el director de projecte Jordi Duran i Cals, vam idear un esquema a mà alçada amb llapis sobre paper reciclat sense renunciar a res i sense pensar en les possibilitats de portar-ho a terme.

Ara, un cop el prototip ja està llest per ser presentat, estic satisfet d'haver estat capaç de portar a terme el projecte tal i com havíem imaginat en un principi.

A nivell tècnic he adquirit moltes habilitats començant pel llenguatge Ruby on Rails que considero un punt molt important al tractar-se d'un llenguatge innovador que no s'imparteix a la carrera, com molts d'altres. Amb això no qüestiono els meus estudis, tot al contrari, simplement que no hi ha prous assignatures per cobrir la varietat de llenguatges existents i evidentment s'ha d'aprendre primer a programar en C igual que primer s'ha d'aprendre a caminar abans de córrer.

Com a elecció personal, m'agraden molt més els llenguatges d'alt nivell orientats a objecte tals com Ruby, Python, Java, etc. i en gaudeixo molt més treballant-t'hi que no pas amb llenguatges procedurals dels de "*picar pedra*". M'agrada pensar que dins del nostre ram com a informàtics també hi ha espai per la creativitat i aquesta és la filosofia de llenguatges com Ruby que premien l'optimització de codi, la simplicitat, reutilització, encapsulament, modularitat, que proporciona un sistema automàtic de reserva i destrucció de memòria, immens ventall de llibreries, orientat a entorns web distribuïts, que mira cap al futur, llenguatge en constatat evolució i en definitiva llibertat absoluta per fer i desfer a voluntat del programador.

També he adquirit altres habilitats tals com treballar en un sistema de màquina virtual amb el VMware Player, millorar el meu coneixement del sistema operatiu Ubuntu, repassar i aplicar la teoria de llenguatges formals impartida a la universitat, instal·lar i configurar un proveïdor de correu electrònic en local, usar el meu portàtil com a servidor web, teoria de protocols TCP/IP, refrescar conceptes de bases de dades, treballar en detall amb phpMyAdmin i MySQL, aprofundir en les eines ofimàtiques, etc.

D'altra banda també he millorat en les habilitats transversals ja que he après a enfrontar-me a una planificació, formalitzar tots els tràmits previs al projecte necessaris per portar-lo a terme i ser-ne responsable justificant cada pas. També he après a autogestionar-me el temps i les activitats, sent jo desenvolupador, analista i assumint tasques de cap de projecte en alguns casos.

Com a conclusió final val a dir que hi hem treballat fort i hem obtingut resultats molt positius.

6. Glossari

6.1. Acrònims i abreviacions

Al llarg d'aquest document s'usen alguns acrònims, abreviacions i algunes paraules d'origen estranger en referència a conceptes tècnics o sobre estàndards. Veure la següent taula.

Acrònim / Abreviació	Nom complet	Significat
AR	Active Record	És la forma en què Ruby on Rails tracta les base de dades fent un paral·lelisme entre la POO i les BD. Les taules són classes, les files són objectes de les classes i les columnes de les taules són els atributs de la classe.
BBDD ó BD	Base de Dades	Conjunt de dades d'un mateix context arxivades d'una forma estricta i ordenada seguint uns criteris comuns.
DBA	Database administrator	Usuari tècnic administrador de la base de dades.
Framework	Capa de desenvolupament web.	Acostuma a ser un afegit a una base de llenguatge de programació o a un programa. Això ens permet importar llibreries i funcionalitats extres que ens faciliten la programació i ens proveeixen solucions. En el cas que ens ocupa, el framework de Rails per a Ruby ens ofereix un conjunt de llibreries, estàndards i aplicacions per a un millor desenvolupament i adaptabilitat al web del nostre programa fet en Ruby.
Freeware	Programari lliure.	El programari catalogat com a tal, té llicència gratuïta per a qualsevol usuari i aquest és lliure per a instal·lar-lo i/o distribuir-lo al nivell de les seves necessitats.
GPL	General Public License	Programari amb llicència pública general. La còpia i/o distribució etc d'aquest programari és permesa.
Hardware	Maquinari	Tot el conjunt de peces físiques que conformen un equip informàtic.
JSON	JavaScript Object Notation	És un llenguatge estructurat per emmagatzemar i compartir informació a gran escala. S'usa bàsicament per entorns web
MVC	Model View Controller	És un paradigma de programació orientat a web.

		Parteix de tres components; el model conté l'estructura de taules i les relacions entre elles, la vista conté el codi html final a través del qual l'usuari veurà les dades i finalment el controlador gestiona els fluxos de dades i les vistes a mostrar en cada moment.
PLN	Processament de Llenguatges Naturals	Interpretar el significat del llenguatge natural usant la computació.
POO	Programació Orientada a Objecte	Paradigma de programació basat en objectes de tipus abstracte de dades. És a dir no es basa en el programes clàssics d'execució procedural, si no que dissenyem varies classes amb els seus mètodes (<i>variables i funcions</i>) i interactuem amb les seves instàncies.
RoR	Ruby on Rails	Llenguatge de programació Ruby amb l'afegit del framework de Rails.
Ruby	-	Llenguatge de programació de 4rta generació interpretat, reflexiu i orientat a objectes. Va ser dissenyat per Yukihiro Matsumoto l'any 1995 i es va inspirar en la semàntica d'altres llenguatges com Perl, Smalltalk, Python i Eiffel.
SO	Sistema Operatiu	Conjunt de programari encarregat de gestionar les accions dels usuaris a través dels perifèrics associats. Alguns exemples serien: MS-DOS, Unix, GNU/Linux, MAC OS, FreeBSD, Windows.
SQL	Structured Query Language	Llenguatge estructurat de consulta a base de dades que el recolza l'estàndard SQL92.
Software	Programari	Programa, o conjunt de programes informàtics.
WebService	Servei web	Té varies aplicacions, la que usem per al nostre projecte és la possibilitat de proporcionar codi executable a aplicacions externes al servidor a través d'importar-ho en forma de llibreries en el codi de programes en altres equips.
XML	Extensible Markup Language	És un metallenguatge extensible de programació en forma d'etiquetes molt estandarditzat. S'usa per estructurar i emmagatzemar grans volums de dades i com a complement per a altres llenguatges de programació.

Taula 48: Relació dels acrònims usats al llarg del document.

6.2. Relació de taules

Taula 1: Taula de catalogació d'objectius.....	13
Taula 2: Stakeholders que intervenen.....	14
Taula 3: Relació dels diferents perfils d'usuari.....	14
Taula 4: Relació de l'equip de projecte.....	15
Taula 5: Resum de la situació actual envers la proposta.....	18
Taula 6: Maquinari disponible a l'empresa.....	20
Taula 7: Personal del sistema.....	21
Taula 8: Sistemes actuals envers proposta.....	22
Taula 9: Relació de normatives a complir i les seves fonts.....	23
Taula 10: Taula dels requisits funcionals.....	24
Taula 11: Requisits no funcionals.....	25
Taula 12: Relació d'objectius de projecte amb requisits funcionals i no funcionals.....	26
Taula 13: Relació d'avantatges i inconvenients entre alternativa 1 i la nostra proposta.....	27
Taula 14: Alternativa 2 envers el prototip.....	28
Taula 15: Calendari de treball de RIS.....	30
Taula 16: Calendari de participació al projecte de JDC.....	30
Taula 17: Calendari de l'adjunt JPC.....	31
Taula 18: Informació dels actors dins del desenvolupament del projecte.....	31
Taula 19: Relació i detalls del maquinari disponible.....	32
Taula 20: Programari disponible.....	33
Taula 21: Taula de les tasques del projecte planificades al Novembre de 2009.....	36
Taula 22: Relació de riscos.....	38
Taula 23: Catalogació dels riscos.....	38
Taula 24: Resultat d'aplicar les solucions als riscos comentats anteriorment.....	39
Taula 25: Justificació costos de personal.....	40
Taula 26: Desgloçament costos de maquinari.....	41
Taula 27: Justificació del cost de les activitats.....	43
Taula 28: Justificació de costos extraordinaris.....	44
Taula 29: Resum de pros i contres de portar a terme el projecte.....	45
Taula 30: Propietats de la taula users.....	57
Taula 31: Propietats de la taula queries.....	58
Taula 32: Taula de la funcionalitat de login.....	75
Taula 33: Taula resum de la funcionalitat de log off.....	76
Taula 34: Funció de llistar usuaris.....	77
Taula 35: Funció de crear usuari.....	77
Taula 36: Detalls de l'acció de crear usuari.....	78
Taula 37: Acció d'eliminar usuari.....	79
Taula 38: Detalls de l'acció de llistar consultes.....	80
Taula 39: Detalls de crear consulta.....	80
Taula 40: Detalls de l'acció de modificar consulta.....	81
Taula 41: Propietats de l'acció d'eliminar consulta.....	82
Taula 42: Taula de proves sobre l'analitzador de llenguatge.....	83
Taula 43: Relació de les proves aplicades sobre la BD.....	84
Taula 44: Detalls de les proves aplicades sobre el site web.....	86
Taula 45: Proves d'unitat sobre el gestor de consultes.....	87
Taula 46: Relació de problemes i solucions aplicades.....	93
Taula 47: Relació de temps i costos addicionals debuts a demora en el desenvolupament del projecte.....	95
Taula 48: Relació dels acrònims usats al llarg del document.....	101

6.3. Relació de figures i il·lustracions

Il·lustració 1: Gràfica representativa de l'estudi del cost/temps treballant amb les eines actuals o amb la nostra proposta.	17
Il·lustració 2: Graf esquemàtic il·lustrant el flux de treball dins de la situació actual.	19
Il·lustració 3: Esquema del maquinari disponible dins del context real.	20
Il·lustració 4: Diagrama de Gantt de la planificació del projecte feta al Novembre.	37
Il·lustració 5: Model de programació web MVC.	47
Il·lustració 6: Gràfic de comparació de diferents aspectes entre PHP, Java i Rails.	48
Il·lustració 7: Diagrama de tot el sistema.	50
Il·lustració 8: Diagrama de flux del procés d'anàlisi de les consultes dels usuaris en llenguatge natural.	52
Il·lustració 9: Diagrama d'estats de l'exemple de gramàtica treetop.	55
Il·lustració 10: Esquema relacional entre les taules del sistema.	59
Il·lustració 11: Model relacional de les taules de mostra usades com a client.	60
Il·lustració 12: Diagrama d'estats del mòdul d'anàlisi i generació de la gramàtica.	61
Il·lustració 13: Logotip del producte.	64
Il·lustració 14: Imatge de la pantalla de benvinguda de l'aplicació d'un usuari qualsevol amb les parts actives marcades.	65
Il·lustració 15: Pàgina de log in pels usuaris.	66
Il·lustració 16: Pàgina principal amb usuari connectat.	66
Il·lustració 17: Vista de la llista d'usuaris.	67
Il·lustració 18: Formulari d'edició de les propietats d'usuari.	67
Il·lustració 19: Vista del llistat de consultes.	68
Il·lustració 20: Vista dels detalls d'una consulta.	68
Il·lustració 21: Vista de crear una consulta nova.	69
Il·lustració 22: Arbre de pàgines de l'aplicació web.	69
Il·lustració 23: Cas d'Ús de tots els usuaris dins de la pàgina principal del site web.	72
Il·lustració 24: CU de l'usuari administrador a la vista usuaris.	73
Il·lustració 25: CU d'usuari sense privilegis a la vista d'usuaris.	73
Il·lustració 26: CU d'administrador a la vista de consultes.	74
Il·lustració 27: CU d'usuari senzill a la interfície de consultes.	74
Il·lustració 28: Diagrama de Seqüència de log in d'usuari.	75
Il·lustració 29: Diagrama de seqüència log off d'usuari.	76
Il·lustració 30: DS Llistar usuaris.	77
Il·lustració 31: DS crear usuari.	78
Il·lustració 32: DS modificar usuari.	78
Il·lustració 33: DS Eliminar usuari.	79
Il·lustració 34: DS Llistar consultes.	80
Il·lustració 35: DS Crear consulta.	81
Il·lustració 36: DS Modificar consulta.	81
Il·lustració 37: DS Eliminar consulta.	82
Il·lustració 38: Diagrama de Gantt del projecte un cop desenvolupat.	97

7. Bibliografia

7.1. Escrita

7.1.1. Apunts.

- Apunts i dossier del curs *S1: Programació en Ruby i Ruby on Rails* de la *Universitat d'Estiu de l'Autònoma 2009*. Amb els professors Abraham Martín i Carles Martínez que es troben actualment impartint classes a la universitat de Cambridge a Anglaterra.
- Apunts i materials de l'assignatura *Enginyeria del Software I* cursada al primer semestre de 2008 a la *Escola Universitària d'Informàtica de Sabadell*.
- Apunts i materials de l'assignatura de *Metodologia i Gestió de Projectes* cursada al primer semestre de 2009 a la EUIS.
- Apunts i materials de l'assignatura *Teoria d'Autòmats i Llenguatges Formals* cursada de Setembre de 2008 a Juny de 2009 també a la EUIS.

7.1.2. Llibres.

- *Agile Web Development with Rails* (3a edició) de **Sam Ruby, Dave Thomas, David Heinemeier Hansson** i múltiples col·laboracions externes, editat per *The Pragmatic Programmers* al març del 2009 a North Carolina, Estats Units d'Amèrica.
- *FXRuby. Create Lean and Mean GUIs with Ruby* (1a edició) de **Lyle Johnson** editat per *The Pragmatic Programmers* al març del 2008 a North Carolina, Estats Units d'Amèrica.
- *Ruby on Rails* (1a edició traduïda) de **Bruce A. Tate** i **Curt Hibbs** editat per *Anaya Multimedia O'REILLY* al 2007 a Madrid, Espanya.

7.2. Digital

- Accenture SL, http://www.accenture.com/Countries/Spain/About_Accenture, 2010.
- Agència Catalana de Protecció de Dades, <http://www.apdcat.net/ca/index.php>, 2009.
- Documentació llibreria estàndard llenguatge de programació Ruby on Rails, <http://stdlib.rubyonrails.org/>, 2010.
- English Query 2000, <http://technet.microsoft.com/en-us/library/aa224772%28SQL.80%29.aspx>, 2009.
- GNOME Planner, <http://live.gnome.org/Planner/Screenshots>, 2010.
- IEEE Std 830-1998, http://www.unap.cl/metadot/index.pl?id=20061&isa=Item&field_name=item_attachment_file&op=download_file, 1998.
- ISO9000, http://www.iso.org/iso/iso_catalogue/management_standards/iso_9000_iso_14000/iso_9000_essentials.htm, 2009.
- ISO20000, http://www.iso.org/iso/catalogue_detail?csnumber=41332, 2009.
- LOPD, https://www.agpd.es/portalwebAGPD/canaldocumentacion/common/estandares_resolucion_madrid.pdf, 2009.
- MySQL, www.mysql.com, 2010.
- Normativa PFC UAB, <http://www.uab.cat/Document/330/254/projectes-normativa20090630.pdf>, 2009.
- Oracle, www.oracle.com, 2010.
- phpMyAdmin, http://www.phpmyadmin.net/home_page/index.php, 2010.
- PHP vs Java vs Rails, <http://www.cmswire.com/cms/industry-news/php-vs-java-vs-ruby-000887.php>, 2010.
- Pragmatic Programmers, <http://pragprog.com/>, 2010.
- SQL-HAL, <http://www.csse.monash.edu.au/hons/projects/2000/Supun.Ruwanpura/>, 2010.
- Subversion, <http://subversion.tigris.org/>, 2010.
- Ubuntu, <http://www.ubuntu.com/>, 2010.
- Ubuntu España, <http://www.ubuntu-es.org/>, 2010.
- VMware Player, <http://www.vmware.com/products/player/overview.html>, 2010.
- Wikipedia, <http://www.wikipedia.org/>, 2010.
- XML, <http://www.w3.org/XML/>, 2010.

8. Agraïments

A la família que sempre ha estat al meu costat fins i tot quan jo estava molt lluny.

Als amics que em van ajudar a concentrar-me en els moments difícils i als companys que em van donar un cop de mà pràctic en temes del projecte amb menció especial al ja diplomat Albert Farràs i Canals.

Al professor i director del projecte Jordi Duran i Cals perquè sempre ha cregut en el projecte i en les meves capacitats per portar-lo a terme.

A la consultoria *Accenture SL* per donar-me l'oportunitat de treballar-hi durant més de 18 mesos sent una de les millors empreses del sector on em va sorgir la idea d'aquest projecte.

Autor del document:
Raimon Izard Serrano
Juny de 2010.